



(19)

(11) Publication number:

09162720 A

Generated Document.

PATENT ABSTRACTS OF JAPAN

(21) Application number: 08097070

(51) Intl. Cl.: H03K 19/0175 G06F 17/50 H03K 19/173

(22) Application date: 18.04.96

(30) Priority: 19.04.95 US 95 423943

(43) Date of application publication: 20.06.97

(84) Designated contracting states:

(71) Applicant: TOSHIBA CORP

(72) Inventor: USAMI MASAYOSHI

(74) Representative:

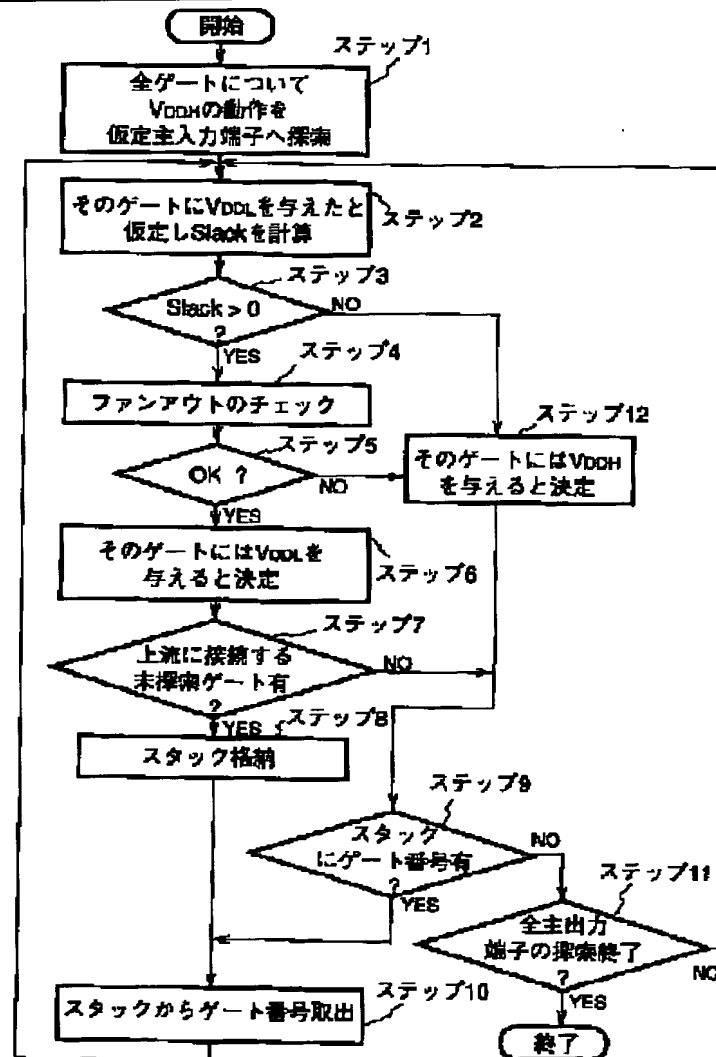
(54) COMBINED LOGIC CIRCUIT AND DESIGNING METHOD FOR THE SAME

(57) Abstract:

PROBLEM TO BE SOLVED: To reduce power consumption by clustering gates constituting a logic circuit based on the difference of their operating voltages and connecting the cluster, which is to be operated by the lowest operating voltage, to the side close to a main output terminal.

SOLUTION: Assuming that all the logic gates are operated by a high operating voltage VDDH, first of all, searching is performed from one main output terminal to a main input terminal. A timing margin Slack is found on the assumption that a low operating voltage VDDL is applied at the met gate (step 2). In the case of $Slack > 0$, fan-out is checked and in the case of $Slack > 0$ even after the VDDL is applied to all the downstream gates, the low operating voltage VDDL is applied to that gate. Next, when an unsearched gate to be connected to that gate exists, all their gate numbers are piled up at a stack. The gates to be checked are successively extracted from the stack, processing is returned to the step 2 and similar processing is performed. This is repeated until the unsearched gates are eliminated and when these gates are eliminated, similar processing is performed from the next output terminal.

COPYRIGHT: (C)1997,JPO



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-162720

(43)公開日 平成9年(1997)6月20日

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
H 0 3 K 19/0175			H 0 3 K 19/00	1 0 1 A
G 0 6 F 17/50		9199-5K	19/173	1 0 1
H 0 3 K 19/173	1 0 1		G 0 6 F 15/60	6 5 6 Z

審査請求 未請求 請求項の数17 O L (全 28 頁)

(21)出願番号 特願平8-97070

(22)出願日 平成8年(1996)4月18日

(31)優先権主張番号 08/423, 943

(32)優先日 1995年4月19日

(33)優先権主張国 米国 (U S)

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 宇佐美 公良

神奈川県川崎市幸区堀川町580番1号 株式会社東芝半導体システム技術センター内

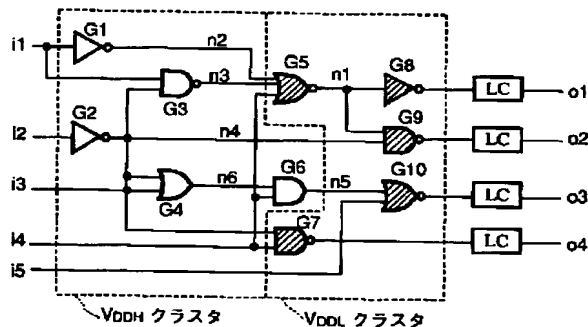
(74)代理人 弁理士 三好 秀和 (外3名)

(54)【発明の名称】 組み合わせ論理回路及びその設計方法

(57)【要約】

【課題】 大規模半導体集積回路で用いられる消費電力の小さな組み合わせ論理回路及びその設計方法を提供する。

【解決手段】 主入力端子と主出力端子を持つ論理回路の消費電力を低減する技術が示されている。この論理回路を構成するゲートは、その動作電圧の違いによってクラスタ化される。まず、主入力端子側には最も高い動作電圧で動かすゲートがクラスタ化し、さらに、そのクラスタの出力に、その次に高い動作電圧で動かすゲートをクラスタ化して接続し、以下同様に次々に接続して、最後に、最も主出力端子に近い側に最も低い動作電圧で動かすゲートをクラスタ化して接続される。



【特許請求の範囲】

【請求項1】 少なくとも1つの主入力端子と少なくとも1つの主出力端子を備えた組み合わせ論理回路であって、入力ノードと出力ノードを持ち第1の動作電圧で駆動される複数の第1のゲートと、入力ノードと出力ノードを持ち前記第1の動作電圧よりも低い第2の動作電圧で駆動される複数の第2のゲートとを備え、前記第2のゲートのすべての出力ノードは、前記第2のゲートの入力ノードと前記主出力端子のいずれかのみに接続されていることを特徴とする組み合わせ論理回路。

【請求項2】 前記第2のゲートの少なくとも1つに接続された前記主出力端子と、前記少なくとも1つの第2のゲートの間にレベルコンバータが設けられていることを特徴とする請求項1に記載の組み合わせ論理回路。

【請求項3】 前記レベルコンバータは、入力した信号を保持し、前記第2の動作電圧から前記第1の動作電圧へ変換するレベル変換機能付ラッチであることを特徴とする請求項1に記載の組み合わせ論理回路。

【請求項4】 前記レベル変換機能付ラッチは、動作電圧VDDHで動作し、互いの入出力ノードを逆方向に接続した一対のインバータからなるラッチ回路を備え、インバータの一方は、第1のトランジスタによって接地レベルに接続され、他方は第2のトランジスタによって接地レベルに接続され、前記第2のトランジスタの入力にはインバータが設けられており、前記第1及び第2のトランジスタは、逆相でオンオフすることを特徴とする請求項3に記載の組み合わせ論理回路。

【請求項5】 前記第1の動作電圧で駆動される複数の第1のゲートと前記第2の動作電圧で駆動される複数の第2のゲートは、夫々別々に平行な複数の列に配置されていることを特徴とする請求項1に記載の組み合わせ論理回路。

【請求項6】 更に、レベルコンバータからなる別の列が前記複数の列と平行に配置されていることを特徴とする請求項5に記載の組み合わせ論理回路。

【請求項7】 前記第1の動作電圧で駆動される複数の第1のゲートと前記第2の動作電圧で駆動される複数の第2のゲートは、それらが互いに接続して構成される複数の信号経路が前記複数の列に対して直角になるように配置されていることを特徴とする請求項5に記載の組み合わせ論理回路。

【請求項8】 少なくとも1つの主入力端子と少なくとも1つの主出力端子を備えた組み合わせ論理回路であって、入力ノードと出力ノードを持ち第1の動作電圧で駆動される複数の第1のゲートと、入力ノードと出力ノードを持ち前記第1の動作電圧よりも低い第2の動作電圧で駆動される複数の第2のゲートと、入力ノードと出力ノードを持ち前記第2の動作電圧よりも低い第3の動作電圧で駆動される複数の第3のゲートとを備え、前記第3のゲートのすべての出力ノードは、前記第3のゲート

の入力ノード及び前記主出力端子のいずれかのみに接続されており、前記第2のゲートのすべての出力ノードは、前記第2のゲートの入力ノード、前記第3のゲートの入力ノード及び前記主出力端子のいずれかのみに接続されていることを特徴とする組み合わせ論理回路。

【請求項9】 少なくとも1つの主入力端子と、少なくとも1つの主出力端子と、それらの間に接続された論理ゲートを備えた組み合わせ論理回路を設計する方法であって、

前記論理ゲートの全体に第1の動作電圧を供給した場合に所定のタイミング・スペックを満足する論理回路を設計する段階と、

前記主出力端子から前記主入力端子へ向かって、前記論理ゲートの夫々について1つずつ前記第1の動作電圧を、前記第1の動作電圧よりも低い第2の動作電圧に変更できるか否かを検討する段階を含み、

前記検討段階では、前記論理ゲートの1つへ前記第2の動作電圧を与えた場合に、前記論理回路全体で前記タイミング・スペックを満たせばその論理ゲートへ供給する電圧を前記第2の動作電圧に決定され、前記タイミング・スペックを満たさなければその論理ゲート及びその入力端子に接続されている論理ゲートへ供給する電圧を前記第1の動作電圧に決定することを特徴とする組み合わせ論理回路を設計する方法。

【請求項10】 前記主出力端子は複数あり、前記主出力端子を1つずつ順に選択し、選択された前記主出力端子に接続されている論理ゲートに関して前記検討段階を行い、それが終了した後、次に選択した前記主出力端子移ることを特徴とする請求項9に記載の組み合わせ論理回路を設計する方法。

【請求項11】 少なくとも1つの主入力端子と、少なくとも1つの主出力端子と、それらの間に接続された論理ゲートを備えた組み合わせ論理回路を設計する方法であって、

前記論理ゲートの全体に第1の動作電圧を供給した場合に所定のタイミング・スペックを満足する論理回路を設計する段階と、

前記主出力端子に対する接続関係の近いゲートから順に、前記論理ゲートの夫々について1つずつ前記第1の動作電圧を、前記第1の動作電圧よりも低い第2の動作電圧に変更できるか否かを検討する段階を含み、

前記検討段階では、前記論理ゲートの1つへ前記第2の動作電圧を与えた場合に、前記論理回路全体で前記タイミング・スペックを満たせばその論理ゲートへ供給する電圧を前記第2の動作電圧に決定され、前記タイミング・スペックを満たさなければその論理ゲート及びその入力端子に接続されている論理ゲートへ供給する電圧を前記第1の動作電圧に決定することを特徴とする組み合わせ論理回路を設計する方法。

【請求項12】 前記主出力端子は複数あり、夫々の主

出力端子にN個(Nは0に初期化)のゲートを介して接続されている論理ゲートに関して前記検討段階を行い、それが終了した後、Nを1ずつ増加させて同じ手順繰り返すことを特徴とする請求項11に記載の組み合わせ論理回路を設計する方法。

【請求項13】 少なくとも1つ以上の主入力端子と、少なくとも1つ以上の主出力端子と、を備えた組み合わせ論理回路であって、入力ノードと出力ノードを持ち第1の動作電圧で駆動される複数の第1のゲートと、入力ノードと出力ノードを持ち前記第1の動作電圧よりも低い第2の動作電圧で駆動される複数の第2のゲートとを備え、前記第2のゲートのすべての出力ノードは、前記第2のゲートの入力ノードと前記主出力端子のいずれかのみに接続されていることを特徴とする組み合わせ論理回路。

【請求項14】 少なくとも1つ以上の主入力端子と、少なくとも1つ以上の主出力端子と、入力ノードと出力ノードを持ち第1の動作電圧で駆動される複数の第1のゲートと、入力ノードと出力ノードを持ち前記第1の動作電圧よりも低い第2の動作電圧で駆動される複数の第2のゲートとを備え、前記主入力端子は前記第1のゲートの入力ノードまたは前記第2のゲートの入力ノードに接続され、前記第1のゲートの出力ノードはその第1のゲートとは異なる前記第1のゲートの入力ノード、前記第2のゲートの入力ノード、前記主出力端子のうちのいずれかに接続され、前記第2のゲートの出力ノードはその第2のゲートとは異なる前記第2のゲートの入力ノードまたはレベルコンバータに接続され、前記レベルコンバータは前記主出力端子に接続されていることを特徴とする組み合わせ論理回路。

【請求項15】 少なくとも1つ以上の主入力端子と、少なくとも1つ以上の主出力端子と、入力ノードと出力ノードを持ち第1の動作電圧で駆動される複数の第1のゲートと、入力ノードと出力ノードを持ち前記第1の動作電圧よりも低い第2の動作電圧で駆動される複数の第2のゲートと、入力ノードと出力ノードを持ち前記第2の動作電圧よりも低い第3の動作電圧で駆動される複数の第3のゲートとを備え、前記主入力端子は前記第1のゲートの入力ノード、前記第2のゲートの入力ノード、前記第3のゲートの入力ノードのうちのいずれかに接続され、前記第1のゲートの出力ノードはその第1のゲートとは異なる前記第1のゲートの入力ノード、前記第2のゲートの入力ノード、前記第3のゲートの入力ノード、前記主出力端子のうちのいずれかに接続され、

前記第2のゲートの出力ノードはその第2のゲートとは異なる前記第2のゲートの入力ノード、前記第3のゲートの入力ノード、レベルコンバータのうちのいずれかに接続され、

前記第3のゲートの出力ノードはその第3のゲートとは異なる前記第3のゲートの入力ノードまたはレベルコンバータに接続され、

前記レベルコンバータは前記主出力端子に接続されていることを特徴とする組み合わせ論理回路。

【請求項16】 第1の動作電圧で駆動される複数の論理回路で構成される第1の領域と、前記第1の動作電圧よりも低い第2の動作電圧で駆動される複数の論理回路で構成される第2の領域と、この第2の領域の出力側の端子が接続されているレベルコンバータとがらなる組み合わせ論理回路を内部に有していることを特徴とする半導体集積回路。

【請求項17】 前記第1の領域の出力側の端子の少なくとも一部は前記第2の領域の入力側の端子に接続されていることを特徴とする半導体集積回路。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、大規模半導体集積回路の消費電力を低減する技術に関する。特に、本発明は、大規模半導体集積回路で用いられる消費電力の小さな組み合わせ論理回路及びその設計方法に関するものである。

【0002】

【従来の技術】CMOS回路の消費電力は、大部分が負荷の充放電によるダイナミック・パワーであり、これは印加されている動作電圧の2乗に比例する。消費電力を低減化する方法として、動作電圧を下げる方法はきわめて効果的であるが、動作電圧を下げるとトランジスタのドレイン電流が減少するため、回路の遅延時間が増大し、性能低下を引き起こす。

【0003】たとえば、図22に示すような組み合わせ論理回路を低消費電力化する際に、各ゲートの動作電圧をすべて下げてしまうと、もともとタイミング・クリティカルパスだったところは、タイミング・スペックを満足できなくなる。

【0004】いっぽう、論理回路の中で、すべてのパスがクリティカルパスになっているわけではない。そこで、クリティカルパスに乗っていない部分のゲートに対してのみ、動作電圧を下げるのが可能である。ところが、CMOS回路の場合には、以下に説明するように、低い動作電圧で動かすゲートと高い動作電圧で動かすゲートのインターフェース部分に、DC電流防止用のレベルコンバータ回路が必要となる。

【0005】ここで、レベルコンバータが何故必要なのかについて、説明を加える。CMOS論理回路中で、2つの異なる動作電圧を使用する場合、図23に示すよう

に、低動作電圧（動作電圧VDDL）で動かす回路と高動作電圧（動作電圧VDDH）で動かす回路をダイレクトに接続すると、インターフェース部分で、DC電流が流れてしまう。

【0006】この理由は、図中のノードN1がハイレベル（動作電圧VDDH）のとき、 V_{thp} をPチャネルトランジスタMP1のしきい値として、もし動作電圧VDDLが動作電圧VDDL<動作電圧VDDH- V_{thp} を満たす程度に低い場合、PチャネルトランジスタMP1は完全にオフせず、NチャネルトランジスタMN1を介して、動作電圧VDDHからVSSに向かってDC電流が流れる。DC電流が流れることによる電力消費は非常に大きいので、通常、DC電流が流れないようにする回路（レベルコンバータ）をインターフェース部分に挟む方法をとる。

【0007】典型的なレベルコンバータ回路を図24に示す。このレベルコンバータ回路は、一方のドレインが他方のゲートに接続された一対のPチャネルトランジスタMP2、MP3からなっている。PチャネルトランジスタMP2は、NチャネルトランジスタMN2によって接地レベルに接続され、他方のPチャネルトランジスタMP3は、第2のNチャネルトランジスタMN5によって接地レベルに接続されている。これらNチャネルトランジスタMN2、MN3は、動作電圧VDDLで動作し、やはり動作電圧VDDLで動作するゲートの出力信号によってオンオフするが、NチャネルトランジスタMN3にはインバータが設けられているので、一方がオンしたときは他方がオフする。

【0008】なお、動作電圧VDDHで動かす回路の出力に、動作電圧VDDLで動かす回路を直接接続しても、こちらのほうは図23の場合と異なり、DC電流が流れる問題はない。即ち、動作電圧VDDLで動かす回路の入力のハイレベルが動作電圧VDDHまで引き上げられるため、Pチャネルトランジスタは完全にオフする。

【0009】

【発明が解決しようとする課題】しかしながら、レベルコンバータは、DC電流を遮断するはたらきをもつものの、スイッチングの際にかなり大きなダイナミック・パワーを消費する。したがって、レベルコンバータの個数が多くなればなるほど、レベルコンバータ全体で消費するパワーが増え、消費電力低減化の効果を弱めてしまう。

【0010】例えば、図22に与えられた論理回路に対し、タイミング制約を満たしながら、動作電圧VDDHを与えるゲートと動作電圧VDDLを与えるゲートを決定する際に、従来技術では、図25のような構造が生成されてしまう場合があった。図中、ハッチングで区別された動作電圧VDDLゲートの出力と動作電圧VDDHゲートの間に、レベルコンバータが設けられてる。すな

わち、図25に示す構造では、動作電圧VDDLゲートの出力が動作電圧VDDHゲートの入力に入る接続構造をたくさん抱えており、そのため多数のレベルコンバータが必要である。

【0011】レベルコンバータの設置自体、かなり大きなダイナミック・パワーを消費するため、多数のレベルコンバータを挿入する構造は、消費電力を低減化するという目的に合致しない。したがって、これまで論理回路の一部分のゲートに対して、動作電圧を下げて消費電力を下げる試みはなされていなかった。

【0012】本発明は、上述の如き従来の問題点を解決するためになされたもので、その目的は、大規模半導体集積回路の消費電力を低減することである。

【0013】また、その他の目的は、大規模半導体集積回路で用いられる消費電力の小さな組み合わせ論理回路及びその設計方法を提供することである。

【0014】

【課題を解決するための手段】上記目的を達成するために、本発明の組み合わせ論理回路は、少なくとも1つの主入力端子と少なくとも1つの主出力端子を備え、入力ノードと出力ノードを持ち第1の動作電圧で駆動される複数の第1のゲートと、入力ノードと出力ノードを持ち前記第1の動作電圧よりも低い第2の動作電圧で駆動される複数の第2のゲートとを備え、前記第2のゲートのすべての出力ノードは、前記第2のゲートの入力ノードと前記主出力端子のいずれかのみに接続されていることを特徴とする。

【0015】又、本発明の組み合わせ論理回路を設計する方法は、少なくとも1つの主入力端子と、少なくとも1つの主出力端子と、それらの間に接続された論理ゲートを備えた組み合わせ論理回路に適用され、前記論理ゲートの全体に第1の動作電圧を供給した場合に所定のタイミング・スペックを満足する論理回路を設計する段階と、前記主出力端子から前記主入力端子へ向かって、前記論理ゲートの夫々について1つずつ前記第1の動作電圧を、前記第1の動作電圧よりも低い第2の動作電圧に変更できるか否かを検討する段階を含み、前記検討段階では、前記論理ゲートの1つへ前記第2の動作電圧を与えた場合に、前記論理回路全体で前記タイミング・スペックを満たせばその論理ゲートへ供給する電圧を前記第2の動作電圧に決定され、前記タイミング・スペックを満たさなければその論理ゲート及びその入力端子に接続されている論理ゲートへ供給する電圧を前記第1の動作電圧に決定することを特徴とする。

【0016】上述の如き構成によれば、一般的な実施形態では、論理回路のゲートは動作電圧によって、次のようにクラスタ化される。まず、主入力端子側には最も高い動作電圧で動かすゲートがクラスタ化し、さらに、そのクラスタの出力に、その次に高い動作電圧で動かすゲートをクラスタ化して接続し、以下同様に次々に接続し

て、最後に、最も主出力端子に近い側に最も低い動作電圧で動かすゲートをクラスタ化して接続される。本発明の回路構造では、レベルコンバータの個数を、最大でも出力ピンと同数に抑えることができるため、消費電力低減化の効果が非常に大きい。

【0017】

【発明の実施の形態】以下、本発明の一実施形態について、図面を用いながら説明する。

【0018】図1は、本発明を利用して設計されたゲートアレイやスタンダードセル等のASICチップの全体を示す図である。このLSIチップは、大まかに3つの部分からなっている。すなわち、外部回路との接続を行うI/O部、予め回路構造の設計されているメガセルで構成される機能ブロックMC及びMB1-3、そして、ランダムロジック部を構成する機能ブロックRL1-5によってこのLSIチップが構成される。ここで、メガセルMCは、メモリ部分であり、メガセルMB1-3はマクロブロックである。

【0019】この内、本発明による設計方法に適した部分は、ロジックを実現するランダムロジック部RL1-5である。又、マイクロプロセッサやDSP等のフルカスタム・チップに適用する場合には、主にその制御ロジック部分に適用すると効果がある。

【0020】機能ブロックRL1-5は単体としては組合わせ回路であり、図2に例示するように、フリップフロップを介して信号のやり取りを行う。夫々の機能ブロックRLのランダムロジックを構成するゲートは、矢印のように一般に複数の信号バスを形成して接続している。そして、同一機能ブロックであっても、内部の信号バスの遅延時間は異なる。タイミング・スペックを満たす信号バス上の各ゲートについては、供給される動作電圧を下げることでできる可能性がある。本発明は、このように、個々の機能ブロック内部の信号バスを調査して、構成ゲートの一部について、低い動作電圧を与えることにより低消費電力構造を実現するものである。

【0021】図22に示す組み合わせ論理回路を用いて、このような機能ブロックを低消費電力構造に設計する方法を説明する。主入力端子(i1-i5)や主出力端子(o1-o5)において、実際に信号が有効となる時刻を信号到着時刻ATとし、入力信号に応じて主出力端子(o1-o5)等で出力される信号が有効となっていないからならない時刻を信号要求時刻RTとする。即ち、主入力端子でのATと主出力端子でのRTとの差異が、この回路に与えられたタイミング制約である。

【0022】この論理回路全体の性能低下を引き起こすことなく消費電力を減らすためには、与えられたタイミング制約を違反することなく、動作電圧VDDHで動かす回路部分と動作電圧VDDLで動かす回路部分を、図22の論理回路の中で形成する必要がある。さらに、消費電力を最小にするためには、レベルコンバータの個数

を最小にし、かつ、動作電圧VDDLで動かす回路部分ができるだけ多くなるようにする必要がある。その為に、動作電圧VDDLで動作するゲートの後には、動作電圧VDDHで動作するゲートは置かないことを条件として設定する。

【0023】まず、図22に与えられた回路中のすべての論理ゲートが、動作電圧VDDHで動作していると仮定する。そして図22の回路全体に対し、スタティック・パス解析というよく知られた手法を施し、各主出力端子での信号到着時刻ATを計算する。一方、各主出力端子について信号要求時刻RTはタイミング制約として与えられているから、RTとATの差を取ることで、各主出力端子でのタイミング余裕(Slack)が求められる。

【0024】タイミング余裕があれば(すなわちSlackが正の値であれば)、その主出力端子に接続したゲートへの供給電圧を下げることでできる可能性がある。しかし、タイミング余裕がなければ(すなわちSlackがゼロか負の値であれば)、その主出力端子に接続したゲートへの供給電圧を下げることはできない。十分のタイミング余裕のある主出力端子について、以下の処理をおこなう。ここでは説明のため、o1、o2、o3、o4のすべてにおいて、Slackが正であるものと仮定する。o1、o2、o3、o4をひとつずつ順に処理していく。選ぶ順序については経験則に従う。有効な順序の例としては、Slackの大きい順に選ぶ方法、又は接続している負荷容量の大きい順に選ぶ方法等がある。

【0025】ここでは、最初にo1を選んだものとする。o1から回路の接続をたどって、上流に向けて動作電圧VDDL(例えば、3.3V)の採用の可能性を探索していく。ここでは探索に、Depth-First-Searchと呼ばれる、深さ方向にたどっていく方法を使う。別の探索方法に、Breadth-First-Searchもある。これらはグラフ理論に基づくもので、"Horowitz, Ellis, et al.; Fundamentals of Data Structures in C, Computer Science Press, 1994"等に一般的な実装方法の説明がある。

【0026】o1から回路の接続をたどると、ゲートG8に出会う。そこで、ゲートG8に対して動作電圧VDDLを供給した場合の、G8のゲート遅延(D8-動作電圧VDDL)を計算する。G8に対して動作電圧VDDHが与えられていたときのゲート遅延(D8-動作電圧VDDH)に比べると、D8-動作電圧VDDLは大きくなる。このD8-動作電圧VDDLの値を用いて、図22の論理回路全体に対し、再度スタティック・パス解析を施す。そして、その結果、o1からo4でのSlackが負にならなければ、G8には動作電圧VDDHを与えることができると判断し、さらに主入力端子に向けて探索を進める。以後、G8のゲート遅延としては、D8-動作電圧VDDLの値を使う。

【0027】次に出会うのはゲートG5であり、今度はG5に対して動作電圧VDDLを供給することを試みる。しかし、G5の出力にはG8だけでなくG9も接続されている。このような場合、先ずG9に動作電圧VDDLを与えることができるかどうかを調べなければならない。ディレイ計算およびスタティック・パス解析の結果、G9及びG5に動作電圧VDDLを与えることができると判断されれば、探索を進めて、次にG1およびG3に動作電圧VDDLが供給できないかを試みるG1とG3のどちらを先にチェックするかについては経験則に従う。

【0028】このようにして、動作電圧VDDLを供給する試みと探索を、Slackが非負であるかぎり続ける。そして、Slackが負となるか、主入力端子に到達したら、次の主出力端子に移り、調査および上記と同様の試みを始める。以上の手続きを全ての主出力端子から行えば、動作電圧VDDL供給の可能性を回路全体のゲートについて調べることができる。

【0029】上記実施形態で述べたとおり、ここでひとつ注意すべきはファンアウトに対する考慮である。即ち、もし調査対象のゲートの出力に複数のゲートが接続されていれば、それら下流のゲートを先に調査しなければならない。たとえば、G8に動作電圧VDDLを供給できることがわかって、次にG5に動作電圧VDDL供給を試みるとする。一方、動作電圧VDDLで動作するゲートの後には、動作電圧VDDHで動作するゲートは置いてはならない。G5の出力には、G8のみならずG9が接続しているから、G5に動作電圧VDDLを与えることができるという前提条件として、G9も動作電圧VDDLを供給ができなければならない。

【0030】このように、与えられた論理回路内の接続構造として、主入力端子→動作電圧VDDHクラスター→動作電圧VDDLクラスター→レベルコンバーター→主出力端子という、図3に示す構造が完結する。その結果、主入力端子から主出力端子への信号の経路としては、次の何れかを取る。

【0031】(1) 主入力端子→動作電圧VDDHが供給されるゲート→動作電圧VDDLが供給されるゲート→レベルコンバーター→主出力端子
または

(2) 主入力端子→動作電圧VDDLが供給されるゲート→レベルコンバーター→主出力端子
または

(3) 主入力端子→動作電圧VDDHが供給されるゲート→主出力端子

上記(3)の場合には、動作電圧VDDLが供給されるゲートが入っていないため、レベルコンバーターは不要となっている。

【0032】上記(1)乃至(3)のいずれの構造にも、「動作電圧VDDLが供給されるゲート→動作電

圧VDDHが供給されるゲート」という接続構造は出現しないため、与えられた論理回路内部で、ゲート間にレベルコンバーターを挿入する必要はない。

【0033】このようにして、すべての主出力端子からの探索および動作電圧VDDL供給の試みが終了すると、最終的に、図3に示すように、図中ハッチングで区別された動作電圧VDDLが供給されるゲート群と、動作電圧VDDHしが供給されるゲート群が、それぞれ出力側と入力側にクラスタ化した回路構造が形成される。与えられた論理回路外部へ出力する際に、0～動作電圧VDDHの電圧振幅が必要なら、動作電圧VDDLが供給されるゲート群(動作電圧VDDLクラスタ)と主出力端子の間に、レベルコンバーターを挿入する。

【0034】以上の処理は、各ゲートについて、図4に示したようなテーブルをメモリ上に作成して行われる。このテーブルには、各エントリに3つの属性が与えられている。すなわち、各ゲートの識別記号に対して、そのゲートが既に探索されているか否かを示すフィールド“Visited”、及びそのゲートに与えられている供給電圧レベル“L/H”が対応づけられている。“Visited”の初期値は“No”、であり、“L/H”の初期値は“H”である。

【0035】以下、本発明による処理手順の詳細を図5のフローチャートを参照しながら説明する。先ず、ステップ1で、すべてのゲートが、動作電圧VDDHで動作していると仮定した上で、1つの主出力端子から主入力端子に向かって探索する。出会ったゲートで、そこに動作電圧VDDLを与えたと仮定し、Slackを計算する(ステップ2)。Slack>0であれば(ステップ3)、ファンアウトをチェックし(ステップ4)、ファンアウトの下流のゲートすべてに動作電圧VDDLを与えたと仮定してもSlack>0であれば(ステップ5)、そのゲートには動作電圧VDDLを与えると決定し(ステップ6)、テーブルの該当するフィールド“Visited”に“Yes”、“L/H”に“L”を書き込んでおく。

【0036】次に、ステップ7で、そのゲートの入力へ接続する未探索(Visited=No)のゲートが存在するかどうかをチェックする。存在する場合、そのゲート番号をすべてスタックに積み(ステップ8)、ステップ10でゲート番号をポップし次にチェックすべきゲートとして設定する。そして、ステップ2に戻って同様の処理を行う。未探索のゲートが存在しなければ、ステップ9でスタックを調べ、スタックに取り出すべきゲート番号があるかどうかを調べ、あればポップし対応するゲートを探索対象に設定する(ステップ10)。存在しなければ、すべての主出力端子からの探索が済んでいるかどうかを調べ(ステップ11)、済んでいれば処理終了、済んでいなければ次の主出力端子からの探索を開始し、最初に出会ったゲートを対象にして、ステップ2に戻り同様の処理を行う。

【0037】もしも、ステップ3でSlack>0でな

ければ、そのゲートには動作電圧VDDHを与えると決定し(ステップ12)、テーブルの該当するフィールド"Visited"に"Yes"、"L/H"に"H"を書き込んでおく。そして、スタックに取り出すべきゲート番号があるかどうかを調べ(ステップ9)、あればポップし対応するゲートを探索対象に設定する(ステップ10)。なければ、すべての主出力端子からの探索が済んでいるかどうかを調べ(ステップ11)、済んでいれば処理終了、済んでいなければ次の主出力端子からの探索を開始し、最初に出会ったゲートを対象にして、ステップ2に戻り同様の処理を行う。

【0038】もしも、ステップ5で、ファンアウトの下流のゲートすべてに動作電圧VDDLを与えることができない場合、そのゲートには動作電圧VDDHを与えると決定し(ステップ12)、テーブルの該当するフィールド"Visited"に"Yes"、"L/H"に"H"を書き込んでおく。そして、スタックに取り出すべきゲート番号があるかどうかを調べ(ステップ9)、あればポップし対応するゲートを探索対象に設定する(ステップ10)。なければ、すべての主出力端子からの探索が済んでいるかどうかを調べ(ステップ11)、済んでいれば処理終了、済んでいなければ次の主出力端子からの探索を開始し、最初に出会ったゲートを対象にして、ステップ2に戻り同様の処理を行う。このようなスタックを用いた処理は、再帰アルゴリズム(recursive algorithm)によるプログラムで実装することができる。

【0039】実際のプログラムでは、ステップ4のファンアウトのチェックは、サブルーチンコールで行う。このサブルーチンに対応するフローチャートを図6に示す。このフローチャートは、図23のステップ4及びステップ5の処理に対応するものである。

【0040】まず、ステップ3でSlack>0であれば、ステップ20で、動作電圧VDDHを与えると決定されたゲート(Visited=Yes, H/L=H)が、そのゲートの出力へ接続しているかどうかをチェックする。接続していなければ、ステップ21で、そのゲートの出力へ接続する未探索(Visited=No)のゲートが存在するかどうかをチェックする。存在する場合、そのゲート番号をすべてスタックに積み(ステップ22)、ステップ23でゲート番号をポップし次にチェックすべきゲートとして設定し、ステップ24でファンアウトリストに登録する。そして、ステップ20に戻って同様の処理を行う。

【0041】ステップ21で、そのゲートの出力へ接続する未探索のゲートが存在しない場合、ステップ25でスタックに取り出すべきゲート番号が残っているかどうかをチェックする。残っていれば、ステップ23でゲート番号をポップし次にチェックすべきゲートとして設定し、ステップ24でファンアウトリストに登録して、ステップ20から同様の処理を行う。ここでいう取り出すべきゲート番号とは、このルーチンでポップしたゲート

番号のことである。つまり、ステップ22でポップしたゲート番号のことである。

【0042】スタックにゲート番号が残っていなければ、ファンアウトリストが完成したので、ステップ26で、このサブルーチンに入る直前のゲートとファンアウトリストに登録されたゲートのすべてに動作電圧VDDLを与える条件で、Slack>0であるかどうかを計算する。

【0043】Slack>0ならば、ステップ27でそのサブルーチンに入る直前のゲートとファンアウトリストに登録されたゲートには動作電圧VDDLを与えると決定し、夫々の"Visited"フィールドに"Yes"、"L/H"に"L"を書き込んで、ファンアウトリストをクリアしておく。そして、メインのルーチンに戻って同様の処理を行う。

【0044】又、Slack>0でないならば、ステップ28でファンアウトリストをクリアして、ステップ12でそのサブルーチンに入る直前のゲートには動作電圧VDDHを与えると決定し、その"Visited"フィールドに"Yes"、"L/H"に"H"を書き込んでおく。そして、メインのルーチンに戻って同様の処理を行う。

【0045】又、ステップ20で、動作電圧VDDHを与えると決定されたゲート(H/M/L=H)が、そのゲートの出力へ接続している場合にも、ステップ28でファンアウトリストをクリアして、ステップ12でそのサブルーチンに入る直前のゲートには動作電圧VDDHを与えると決定し、その"Visited"フィールドに"Yes"、"L/H"に"H"を書き込んで、メインのルーチンに戻って同様の処理を行う。

【0046】以上は、クラスタが2段の場合であるが、更に1つ又は複数の中間の動作電圧を設定しても良い。図8のフローチャートを参照して、3段の場合を説明する。すなわち、高い動作電圧VDDH、中間の動作電圧VDDM及び低い動作電圧VDDLの三種類の動作電圧を与える場合を説明する。

【0047】この場合、各ゲートについて、図7に示したようなテーブルをメモリ上に作成して行われる。このテーブルには、各エントリに3つの属性が与えられている。すなわち、各ゲートの識別記号に対して、そのゲートが既に探索されているか否かを示すフィールド"Visited"、及びそのゲートに与えられている供給電圧レベル"L/M/H"が対応づけられている。"Visited"の初期値は"No"、であり、"L/M/H"の初期値は"H"である。

【0048】まず、ステップ31で、すべてのゲートが、動作電圧VDDHで動作していると仮定した上で、1つの主出力端子から主入力端子に向かって探索する。出会ったゲートで、そこに動作電圧VDDLを与えたと仮定し、Slackを計算する(ステップ32)。Slack>0であれば(ステップ33)、ファンアウトをチェックする。ここでは、先ずステップ34で、動作電

圧VDDHを与えなければならないかどうかをチェックする。動作電圧VDDM又は動作電圧VDDLで動作する可能性があれば、ファンアウトの下流のゲートすべてに動作電圧VDDLを与えたと仮定してもSlack>0であるかどうかチェックする(ステップ35)。Slack>0であれば、そのゲートには動作電圧VDDLを与えると決定し(ステップ36)、テーブルの該当するフィールド"Visited"に"Yes"、"L/M/H"に"L"を書き込んでおく。

【0049】次に、ステップ37で、そのゲートの入力へ接続する未探索(Visited=No)のゲートが存在するかどうかをチェックする。存在する場合、そのゲート番号をすべてスタックに積み(ステップ38)、ステップ40でゲート番号をポップし次にチェックすべきゲートとして設定する。そして、ステップ32に戻って同様の処理を行う。未探索のゲートが存在しなければ、ステップ39でスタックを調べ、スタックに取り出すべきゲート番号があるかどうかを調べ、あればポップし対応するゲートを探索対象に設定する(ステップ40)。存在しなければ、すべての主出力端子からの探索が済んでいるかどうかを調べ(ステップ41)、済んでいれば処理終了、済んでいなければ次の主出力端子からの探索を開始し、最初に出会ったゲートを対象にして、ステップ2に戻り同様の処理を行う。

【0050】もしも、ステップ33でSlack>0でなければ、そのゲートには動作電圧VDDMを与えるか動作電圧VDDHを与えるかを決定しなければならない。その処理を、図9のフローチャートを参照して説明する。

【0051】先ず、対象ゲートに動作電圧VDDMを与えたと仮定し、Slackを計算する(ステップ53)。Slack>0であれば、ファンアウトをチェックし(ステップ54)、ファンアウトの下流のゲートすべてに動作電圧VDDMを与えたと仮定してもSlack>0であれば(ステップ55)、そのゲートには動作電圧VDDMを与えると決定し(ステップ56)、テーブルの該当するフィールド"Visited"に"Yes"、"L/M/H"に"M"を書き込んでおく。

【0052】次に、ステップ57で、そのゲートの入力へ接続する未探索(Visited=No)のゲートが存在するかどうかをチェックする。存在する場合、そのゲート番号をすべてスタックに積み(ステップ58)、ステップ60でゲート番号をポップし次にチェックすべきゲートとして設定する。そして、ステップ52に戻って同様の処理を行う。未探索のゲートが存在しなければ、ステップ59でスタックを調べ、スタックに取り出すべきゲート番号があるかどうかを調べ、あればポップし対応するゲートを探索対象に設定する(ステップ60)。ここでいう取り出すべきゲート番号とは、このルーチンでポップしたゲート番号のことである。つまり、ステップ58で

ポップしたゲート番号のことである。

【0053】もし、ステップ59でスタックに取り出すべきゲート番号が存在しなければ、メインのルーチンのステップ39へ戻り、そこでスタックにゲート番号が残っているかどうかを調べ、残っていればポップし対応するゲートを探索対象に設定する(ステップ40)。ステップ39で取り出すゲート番号とは、このメインのルーチンでポップしたゲート番号となる。つまり、ステップ38でポップしたゲート番号である。その後、メインのルーチンでの処理は、同様に行われる。

【0054】もしも、ステップ53でSlack>0でなければ、そのゲートには動作電圧VDDHを与えると決定し(ステップ60)、テーブルの該当するフィールド"Visited"に"Yes"、"L/M/H"に"H"を書き込んでおく。そして、スタックに取り出すべきゲート番号があるかどうかを調べ(ステップ58)、あればポップし対応するゲートを探索対象に設定する(ステップ59)。なければ、メインのルーチンのステップ39へ戻り、メインのルーチンでの処理が行われる。

【0055】ステップ54のファンアウトのチェックも、サブルーチンコールで行う。このサブルーチンに対応するフローチャートを図10に示す。このフローチャートは、図9のステップ54及びステップ55の処理に対応するものである。

【0056】先ず、ステップ53でSlack>0であれば、ステップ70で、動作電圧VDDHを与えると決定されたゲート(H/M/L=H)が、そのゲートの出力へ接続しているかどうかをチェックする。接続していなければ、ステップ71で、そのゲートの出力へ接続する未探索(Visited=No)のゲートが存在するかどうかをチェックする。存在する場合、そのゲート番号をすべてスタックに積み(ステップ72)、ステップ73でゲート番号をポップし次にチェックすべきゲートとして設定し、ステップ74でファンアウトリストに登録する。そして、ステップ71に戻って同様の処理を行う。

【0057】ステップ71で、そのゲートの出力へ接続する未探索のゲートが存在しない場合、ステップ75でスタックに取り出すべきゲート番号が残っているかどうかをチェックする。残っていれば、ステップ73でゲート番号をポップし次にチェックすべきゲートとして設定し、ステップ74でファンアウトリストに登録して、ステップ70から同様の処理を行う。ここでいう取り出すべきゲート番号とは、このルーチンでポップしたゲート番号のことである。つまり、ステップ72でポップしたゲート番号のことである。

【0058】スタックにゲート番号が残っていなければ、ファンアウトリストが完成したので、ステップ76で、このサブルーチンに入る直前のゲートとファンアウトリストに登録されたゲートのすべてに動作電圧VDDMを与える条件で、Slack>0であるかどうかを計

算する(ステップ95)。

【0059】 $Slack > 0$ ならば、ステップ77でそのサブルーチンに入る直前のゲートとファンアウトリストに登録されたゲートには動作電圧VDDMを与えると決定し、夫々の"Visited"フィールドに"Yes"、"H/M/L"に"M"を書き込んで、ファンアウトリストをクリアしておく。そして、メインのルーチンに戻って同様の処理を行う。

【0060】又、 $Slack > 0$ でないならば、ステップ78でファンアウトリストをクリアして、ステップ62に戻って、動作電圧VDDHを与えると決定する。

【0061】又、ステップ70で、動作電圧VDDHを与えると決定されたゲート(H/M/L=H)が、そのゲートの出力へ接続している場合にも、ステップ78でファンアウトリストをクリアして、動作電圧VDDHを与えると決定する(ステップ62)。

【0062】ステップ34とステップ35の処理は、サブルーチンコールで行う。このサブルーチンに対応するフローチャートを図11に示す。まず、ステップ79で、動作電圧VDDHを与えると決定されたゲート("Visited=Yes"、"H/M/L=H")が、そのゲートの出力へ接続しているかどうかをチェックする。接続していなければ、ステップ80で、動作電圧VDDMを与えると決定されたゲート("Visited=Yes"、"H/M/L=M")が、そのゲートの出力へ接続しているかどうかをチェックする。これも接続していなければ、ステップ81で、そのゲートの出力へ接続する未探索(Visited=No)のゲートが存在するかどうかをチェックする。存在する場合、そのゲート番号をすべてスタックに積み(ステップ82)、ステップ83でゲート番号をポップし次にチェックすべきゲートとして設定し、ステップ84でファンアウトリストに登録する。そして、ステップ79に戻って同様の処理を行う。

【0063】ステップ81で、そのゲートの出力へ接続する未探索のゲートが存在しない場合、ステップ85でスタックに取り出すべきゲート番号が残っているかどうかをチェックする。残っていれば、ステップ83でゲート番号をポップし次にチェックすべきゲートとして設定し、ステップ84でファンアウトリストに登録して、ステップ79から同様の処理を行う。ここでいう取り出すべきゲート番号とは、このルーチンでポップしたゲート番号のことである。つまり、ステップ82でポップしたゲート番号のことである。

【0064】スタックにゲート番号が残っていなければ、ファンアウトリストが完成したので、ステップ86で、このサブルーチンに入る直前のゲートとファンアウトリストに登録されたゲートのすべてに動作電圧VDDLを与える条件で、 $Slack > 0$ であるかどうかを計算する。

【0065】 $Slack > 0$ ならば、ステップ36でそ

のサブルーチンに入る直前のゲートとファンアウトリストに登録されたゲートには動作電圧VDDLを与えると決定し、夫々の"Visited"フィールドに"Yes"、"H/M/L"に"L"を書き込んで、ファンアウトリストをクリアしておく。そして、メインのルーチンに戻って同様の処理を行う。

【0066】又、ステップ79で、動作電圧VDDHを与えると決定されたゲート("Visited=Yes"、"H/M/L=H")が、そのゲートの出力へ接続していれば、最初のゲートの"Visited"フィールドに"Yes"、"H/M/L"に"H"を書き込んで、ファンアウトリストをクリアしておく(ステップ88)。

【0067】又、ステップ80で、動作電圧VDDMを与えると決定されたゲート("Visited=Yes"、"H/M/L=M")が、そのゲートの出力へ接続していれば、図9のステップ52へ移り、その後は動作電圧VDDMが与えられるかどうかを、主入力端子へ向かって探索していく。手順は図9のフローチャートで説明した通り。

【0068】以上は、Depth-First-Searchと呼ばれる方法を応用している。別の探索方法として、Breadth-First-Searchと呼ばれる幅方向にたどっていく方法を利用しても良い。以下、その具体例を示す。

【0069】再び、図22に示す組み合わせ論理回路を用いて、機能ブロックを低消費電力構造に設計する方法を説明する。ここでも、主入力端子(i1-i5)や主出力端子(o1-o5)において、実際に信号が有効となる時刻を信号到着時刻ATとし、入力信号に応じて主出力端子(o1-o5)等で出力される信号が有効となっていないなければならない時刻を信号要求時刻RTとする。

【0070】この論理回路全体の性能低下を引き起こすことなく消費電力を減らすために、動作電圧VDDHで動かす回路部分と、動作電圧VDDM動作電圧で動かす回路部分と、動作電圧VDDLで動かす回路部分を、夫々Breadth-First-Searchに従って図22の論理回路の中に形成する手順を説明する。

【0071】まず、図22に与えられた回路中のすべての論理ゲートが、動作電圧VDDHで動作していると仮定する。そして図22の回路全体に対し、スタティック・パス解析によって各主出力端子での信号到着時刻ATを計算する。一方、各主出力端子について信号要求時刻RTはタイミング制約として与えられているから、RTとATの差を取ることで、各主出力端子でのタイミング余裕(Slack)が求められる。タイミング余裕があれば、その主出力端子に接続した夫々のゲートへの供給電圧を下げるができるかどうかを上流に向かって探索していく。ここまでは、Depth-First-Searchによる上記の方法と同じである。Breadth-First-Searchでは、探索の順序が異なる。

【0072】まず、説明のため、o1、o2、o3、o

4のすべてにおいて、Slackが正であるものと仮定する。o1、o2、o3、o4をひとつずつ順に処理していく。選ぶ順序については経験則に従う。有効な順序の例としては、Slackの大きい順に選ぶ方法、又は接続している負荷容量の大きい順に選ぶ方法等がある。

【0073】ここでは、最初にo1を選んだものとする。Depth-First-Searchでは、o1から回路の接続をたどって、上流に向けて動作電圧VDDLの採用の可能性を探索していくが、Breadth-First-Searchでは、主出力端子o1、o2、o3、o4に近いゲートから探索していく。即ち、主出力端子o1、o2、o3、o4に直接接続しているゲートを0レベル、ゲートを1つ介して接続しているゲートを1レベル、ゲートを2つ介して接続しているゲートを2レベル、ゲートを3つ介して接続しているゲートを3レベル等として、夫々のレベル毎にSlackの計算を行う。

【0074】先ず、o1から回路の接続をたどると、ゲートG8に出会う。そこで、ゲートG8に対して動作電圧VDDLを供給した場合の、G8のゲート遅延(D8-動作電圧VDDL)を計算する。G8に対して動作電圧VDDHが与えられていたときのゲート遅延(D8-動作電圧VDDH)に比べると、D8-動作電圧VDDLは大きくなる。このD8-動作電圧VDDLの値を用いて、図22の論理回路全体に対し、再度スタティック・パス解析を施す。そして、その結果、o1からo4でのSlackが負にならなければ、G8には動作電圧VDDLを与えることができると判断する。

【0075】その後、Breadth-First-Searchでは、主出力端子o2へ移る。o2から回路の接続をたどると、ゲートG9に出会うので、そこで、ゲートG9に対して動作電圧VDDLを供給した場合に、図22の論理回路全体に対し、再度スタティック・パス解析を施してSlackを計算する。そして、主出力端子o3、o4に対しても、同様の処理を行う。その結果、ゲートG8、G9、G10、G7には動作電圧VDDLを与えることができると判断する。

【0076】次に探索するゲートは、主出力端子o1、o2、o3、o4から、ゲートを1つ介して接続しているゲートである。すなわち、ゲートG5、G2、G6を対象にして、動作電圧VDDLを供給した場合の全体のSlackを計算する。尚、主出力端子o4とゲートを1つ介して接続しているゲートはゲートG2であり、主出力端子o2とゲートを1つ介して接続しているゲートと同一である。

【0077】その次に探索するゲートは、主出力端子o1、o2、o3、o4から、ゲートを2つ介して接続しているゲートである。すなわち、ゲートG1、G3、G4を対象にして、動作電圧VDDLを供給した場合の全体のSlackを計算する。

【0078】更に、その次に探索するゲートは、主出力

端子o1、o2、o3、o4から、ゲートを3つ介して接続しているゲートである。この場合、ゲートG1だけが対象である。尚、動作電圧VDDHは与えられず、動作電圧VDDMの供給が決定したゲートの下流では、動作電圧VDDMを与えられるかどうかの調査を行う。

又、動作電圧VDDHの供給が決定したゲートの下流にあるゲートは、動作電圧VDDHの供給が自動的に決定するのはDepth-First-Searchと同じである。

【0079】勿論、各レベルに属するゲートには多数重複があり、ファンアウトの調査も行うので、計算の不要なゲートは飛ばしていく。

【0080】その結果、図12に示したように、動作電圧VDDHで動かす回路部分である動作電圧VDDHクラスと、動作電圧VDDMで動かす回路部分である動作電圧VDDMクラスと、動作電圧VDDLで動かす回路部分である動作電圧VDDLクラスが確定する。

【0081】以上の処理も、Depth-First-Searchと同様に、各ゲートについて、図に示したようなテーブルをメモリ上に作成して行われる。このテーブルには、各エントリに3つの属性が与えられている。すなわち、各ゲートの識別記号に対して、そのゲートが既に探索されているか否かを示すフィールド"Visited"、及びそのゲートに与えられている供給電圧レベル"L/H"が対応づけられている。"Visited"の初期値は"No"、であり、"L/H"の初期値は"H"である。

【0082】以下、詳細を図13のフローチャートを参照しながら説明する。先ず、ステップ101で、すべてのゲートが、動作電圧VDDHで動作していると仮定した上で、レベルを表すレベル番号Nを0で初期化する。ステップ103で、レベル番号Nの未探索のゲートを1つ選択し、そのゲートに動作電圧VDDLを与えたと仮定し(ステップ104)、Slackを計算する(ステップ105)。Slack>0であれば、ファンアウトをチェックし(ステップ106)、ファンアウトの下流のゲートすべてに動作電圧VDDLを与えたと仮定してもSlack>0であれば(ステップ107)、そのゲートには動作電圧VDDLを与えると決定し(ステップ108)、テーブルの該当するフィールド"Visited"に"Yes"、"L/H"に"L"を書き込んでおく。

【0083】次に、ステップ109で、同一レベルの未探索のゲートが存在するかどうかをチェックする。存在する場合、ステップ104に戻って、そのレベルでの探索を続ける。同一レベルの未探索のゲートが存在しない場合、ステップ110で、次のレベル番号N+1のゲートが存在するか否かをチェックする。存在する場合、ステップ103に戻って、レベル番号N+1のゲートについて探索を続ける。レベル番号N+1のゲートが存在しない場合、処理終了。

【0084】もしも、ステップ105でSlack>0でなければ、そのゲートには動作電圧VDDHを与える

と決定し(ステップ112)、テーブルの該当するフィールド "Visited" に "Yes"、"L/H" に "H" を書き込んでおく。そして、同一レベルの未探索のゲートが存在するかどうかを調べ(ステップ109)、あれば、ステップ104に戻って、そのレベルでの探索を続ける。同一レベルの未探索のゲートが存在しない場合は、上記のようにステップ103に戻って、レベル番号N+1のゲートについて探索を続ける。

【0085】もしも、ステップ107で、ファンアウトの下流のゲートすべてに動作電圧VDDLを与えることができない場合、そのゲートには動作電圧VDDHを与えると決定し(ステップ112)、テーブルの該当するフィールド "Visited" に "Yes"、"L/H" に "H" を書き込んでおく。そして、同一レベルの未探索のゲートが存在するかどうかを調べ(ステップ109)、あれば、ステップ104に戻って、そのレベルでの探索を続ける。同一レベルの未探索のゲートが存在しない場合、ステップ110で、次のレベル番号N+1のゲートが存在するか否かをチェックする。存在する場合、ステップ103に戻って、レベル番号N+1のゲートについて探索を続ける。レベル番号N+1のゲートが存在しない場合、処理を終了する。

【0086】実際のプログラムでは、ステップ106のファンアウトのチェックは、サブルーチンコールで行う。このサブルーチンに対応するフローチャートを図14に示す。このフローチャートは、図12のステップ106及びステップ107の処理に対応するものである。

【0087】先ず、ステップ105でSlack>0であれば、ステップ120で、動作電圧VDDHを与えると決定されたゲート(Visited=Yes, H/L=H)が、そのゲートの出力へ接続しているかどうかをチェックする。接続していなければ、ステップ121で、そのゲートの出力へ接続する未探索(Visited=No)のゲートが存在するかどうかをチェックする。存在する場合、そのゲート番号をすべてスタックに積み(ステップ122)、ステップ123でゲート番号をポップし次にチェックすべきゲートとして設定し、ステップ124でファンアウトリストに登録する。そして、ステップ120に戻って同様の処理を行う。

【0088】ステップ121で、そのゲートの出力へ接続する未探索のゲートが存在しない場合、ステップ125でスタックに取り出すべきゲート番号が残っているかどうかをチェックする。残っていれば、ステップ123でゲート番号をポップし次にチェックすべきゲートとして設定し、ステップ124でファンアウトリストに登録して、ステップ120から同様の処理を行う。ここでいう取り出すべきゲート番号とは、このルーチンでポップしたゲート番号のことである。つまり、ステップ122でポップしたゲート番号のことである。

【0089】スタックにゲート番号が残っていなけれ

ば、ファンアウトリストが完成したので、ステップ126で、このサブルーチンに入る直前のゲートとファンアウトリストに登録されたゲートのすべてに動作電圧VDDLを与える条件で、Slack>0であるかどうかを計算する。

【0090】Slack>0ならば、ステップ127でそのサブルーチンに入る直前のゲートとファンアウトリストに登録されたゲートには動作電圧VDDLを与えることと決定し、夫々の"Visited" フィールドに "Yes"、"L/H" に "L" を書き込んで、ファンアウトリストをクリアしておく。そして、メインのルーチンに戻って同様の処理を行う。

【0091】又、Slack>0でないならば、ステップ128でファンアウトリストをクリアして、ステップ112でそのサブルーチンに入る直前のゲートには動作電圧VDDHを与えると決定し、その"Visited" フィールドに "Yes"、"L/H" に "H" を書き込んでおく。そして、メインのルーチンに戻って同様の処理を行う。

【0092】又、ステップ120で、動作電圧VDDHを与えると決定されたゲート(H/M/L=H)が、そのゲートの出力へ接続している場合にも、ステップ128でファンアウトリストをクリアして、ステップ112でそのサブルーチンに入る直前のゲートには動作電圧VDDHを与えると決定し、その"Visited" フィールドに "Yes"、"L/H" に "H" を書き込んで、メインのルーチンに戻って同様の処理を行う。

【0093】以上は、クラスタが2段の場合であるが、Breadth-First-Search を用いた場合でも、更に1つ又は複数の中間の動作電圧を設定することも可能である。図のフローチャートを参照して、3段の場合を説明する。すなわち、高い動作電圧VDDH、中間の動作電圧VDDM及び低い動作電圧VDDLの三種類の動作電圧を与える場合を説明する。

【0094】この場合、各ゲートについて、やはり図7に示したようなテーブルをメモリ上に作成して行われる。このテーブルには、各エントリに3つの属性を与えられている。すなわち、各ゲートの識別記号に対して、そのゲートが既に探索されているか否かを示すフィールド "Visited"、及びそのゲートに与えられている供給電圧レベル "L/M/H" が対応づけられている。"Visited" の初期値は"No"、であり、"L/M/H" の初期値は"H" である。

【0095】以下、処理手順の詳細を図15のフローチャートを参照しながら説明する。先ず、ステップ131で、すべてのゲートが、動作電圧VDDHで動作していると仮定した上で、レベルを表すレベル番号Nを0で初期化する(ステップ132)。ステップ133で、レベル番号Nの未探索のゲートを1つ選択し、そのゲートに動作電圧VDDLを与えたと仮定し、Slackを計算する(ステップ134)。Slack>0であれば(ス

テップ135)、ファンアウトをチェックする。ここでは、先ずステップ136で、動作電圧VDDHを与えなければならないかどうかをチェックする。動作電圧VDDM又は動作電圧VDDLで動作する可能性があれば、ファンアウトの下流のゲートすべてに動作電圧VDDLを与えたと仮定してもSlack>0であるかどうかチェックする(ステップ137)。Slack>0であれば、そのゲートには動作電圧VDDLを与えると決定し(ステップ138)、テーブルの該当するフィールド"Visited"に"Yes"、"L/M/H"に"L"を書き込んでおく。

【0096】次に、ステップ139で、同一レベルの未探索のゲートが存在するかどうかをチェックする。存在する場合、ステップ133に戻って、そのレベルでの探索を続ける。同一レベルの未探索のゲートが存在しない場合、ステップ140で、次のレベル番号N+1のゲートが存在するか否かチェックする。存在する場合、ステップ133に戻って、レベル番号N+1のゲートについて探索を続ける。レベル番号N+1のゲートが存在しない場合は処理を終了する。

【0097】もしも、ステップ135でSlack>0でなければ、そのゲートには動作電圧VDDMを与えるか動作電圧VDDHを与えるかを決定しなければならない。その処理を、図16のフローチャートを参照して説明する。

【0098】先ず、対象ゲートに動作電圧VDDMを与えたと仮定し、Slackを計算する(ステップ152)。Slack>0であれば、ファンアウトをチェックし(ステップ156)、ファンアウトの下流のゲートすべてに動作電圧VDDLを与えたと仮定してもSlack>0であれば(ステップ157)、そのゲートには動作電圧VDDLを与えると決定し(ステップ158)、テーブルの該当するフィールド"Visited"に"Yes"、"L/H"に"L"を書き込んでおく。そして、ステップ139から同様の処理を行う。

【0099】もしも、ステップ155でSlack>0でなければ、そのゲートには動作電圧VDDHを与えると決定し(ステップ112)、テーブルの該当するフィールド"Visited"に"Yes"、"L/H"に"H"を書き込んでおく。そして、ステップ139から同様の処理を行う。

【0100】又、もしも、ステップ157で、ファンアウトの下流のゲートすべてに動作電圧VDDLを与えることができない場合、そのゲートには動作電圧VDDHを与えると決定し(ステップ152)、テーブルの該当するフィールド"Visited"に"Yes"、"L/H"に"H"を書き込んでおく。そして、ステップ139から同様の処理を行う。

【0101】ステップ156のファンアウトのチェックも、サブルーチンコールで行う。このサブルーチンに対

応するフローチャートを図17に示す。このフローチャートは、図16のステップ156及びステップ157の処理に対応するものである。

【0102】先ず、ステップ155でSlack>0であれば、ステップ170で、動作電圧VDDHを与えると決定されたゲート(H/M/L=H)が、そのゲートの出力へ接続しているかどうかをチェックする。接続していなければ、ステップ171で、そのゲートの出力へ接続する未探索(Visited=No)のゲートが存在するかどうかをチェックする。存在する場合、そのゲート番号をすべてスタックに積み(ステップ172)、ステップ173でゲート番号をポップし次にチェックすべきゲートとして設定し、ステップ174でファンアウトリストに登録する。そして、ステップ171に戻って同様の処理を行う。

【0103】ステップ171で、そのゲートの出力へ接続する未探索のゲートが存在しない場合、ステップ175でスタックに取り出すべきゲート番号が残っているかどうかをチェックする。残っていれば、ステップ173でゲート番号をポップし次にチェックすべきゲートとして設定し、ステップ174でファンアウトリストに登録して、ステップ170から同様の処理を行う。ここでいう取り出すべきゲート番号とは、このルーチンでポップしたゲート番号のことである。つまり、ステップ172でポップしたゲート番号のことである。

【0104】スタックにゲート番号が残っていないならば、ファンアウトリストが完成したので、ステップ176で、このサブルーチンに入る直前のゲートとファンアウトリストに登録されたゲートのすべてに動作電圧VDDMを与える条件で、Slack>0であるかどうかを計算する。

【0105】Slack>0ならば、ステップ177でそのサブルーチンに入る直前のゲートとファンアウトリストに登録されたゲートには動作電圧VDDMを与えると決定し、夫々の"Visited"フィールドに"Yes"、"H/M/L"に"M"を書き込んで、ファンアウトリストをクリアしておく。そして、メインのルーチンに戻って同様の処理を行う。

【0106】又、Slack>0でないならば、ステップ178でファンアウトリストをクリアして、ステップ152に戻って、動作電圧VDDHを与えると決定する。

【0107】又、ステップ170で、動作電圧VDDHを与えると決定されたゲート(H/M/L=H)が、そのゲートの出力へ接続している場合にも、ステップ178でファンアウトリストをクリアして、動作電圧VDDHを与えると決定する(ステップ152)。

【0108】ステップ136とステップ137の処理は、サブルーチンコールで行う。このサブルーチンに対応するフローチャートを図18に示す。先ず、ステップ

179で、動作電圧VDDHを与えると決定されたゲート("Visited=Yes"、"H/M/L= H") が、そのゲートの出力へ接続しているかどうかをチェックする。接続していなければ、ステップ180で、動作電圧VDDMを与えると決定されたゲート("Visited=Yes"、"H/M/L= M") が、そのゲートの出力へ接続しているかどうかをチェックする。これも接続していなければ、ステップ181で、そのゲートの出力へ接続する未探索(Visited=No)のゲートが存在するかどうかをチェックする。存在する場合、そのゲート番号をすべてスタックに積み(ステップ182)、ステップ183でゲート番号をポップし次にチェックすべきゲートとして設定し、ステップ184でファンアウトリストに登録する。そして、ステップ179に戻って同様の処理を行う。

【0109】ステップ181で、そのゲートの出力へ接続する未探索のゲートが存在しない場合、ステップ185でスタックに取り出すべきゲート番号が残っているかどうかをチェックする。残っていれば、ステップ183でゲート番号をポップし次にチェックすべきゲートとして設定し、ステップ184でファンアウトリストに登録して、ステップ179から同様の処理を行う。ここでいう取り出すべきゲート番号とは、このルーチンでポップしたゲート番号のことである。つまり、ステップ182でポップしたゲート番号のことである。

【0110】スタックにゲート番号が残っていなければ、ファンアウトリストが完成したので、ステップ186で、このサブルーチンに入る直前のゲートとファンアウトリストに登録されたゲートのすべてに動作電圧VDDLを与える条件で、Slack>0であるかどうかを計算する。

【0111】Slack>0ならば、ステップ136でそのサブルーチンに入る直前のゲートとファンアウトリストに登録されたゲートには動作電圧VDDLを与えると決定し、夫々の"Visited" フィールドに "Yes"、"H/M/L" に "L" を書き込んで、ファンアウトリストをクリアしておく。そして、メインのルーチンに戻って同様の処理を行う。

【0112】又、ステップ179で、動作電圧VDDHを与えると決定されたゲート("Visited=Yes"、"H/M/L= H") が、そのゲートの出力へ接続していれば、最初のゲートの"Visited" フィールドに "Yes"、"H/M/L" に "H" を書き込んで、ファンアウトリストをクリアしておく(ステップ188)。

【0113】又、ステップ180で、動作電圧VDDMを与えると決定されたゲート("Visited=Yes"、"H/M/L= M") が、そのゲートの出力へ接続していれば、図のステップ152へ移り、その後は動作電圧VDDMが与えられるかどうかを、主入力端子へ向かって探索していく。手順は図のフローチャートで説明した通り。

【0114】動作電圧VDDLの値は、本発明を適用す

る回路全体の消費電力が、最小になるように設定するのが最も望ましい。回路全体の消費電力を最小にする動作電圧VDDLの値は、タイミング制約を満たす範囲で、「(VDDLゲート単体での消費電力の削減量) x (VDDLゲートに置換される個数)」が最大になる値である。「VDDLゲート単体での消費電力の削減量」は、動作電圧VDDLが低ければ低いほど大きくなる。一方、「VDDLゲートに置換される個数」は、動作電圧VDDLの電圧値が高ければ高いほど多くなる。これは、動作電圧VDDLが低ければ低いほどVDDLゲートの遅延が大きくなり、タイミング制約を満たす範囲でVDDLゲートへの置換が行いにくくなるためである。以上から、「VDDLゲート単体での消費電力の削減量」と「VDDLゲートに置換される個数」の積は、VDDLが0ボルトからVDDHの間のどこかの点で、最大値をとることがわかる。

【0115】上記の積が最大になるVDDLの値を実際を求めるには、次のようにして行う。動作電圧VDDLの値を、ある刻みで動作電圧VDDHより少しずつ低く設定し、本発明の回路構造を実際に生成して、「(VDDLゲート単体での消費電力の削減量) x (VDDLゲートに置換される個数)」の値を順次求める。そして、前記の積が最大になる動作電圧VDDLの値を見つければよい。経験的には、図19に示したような関係があり、動作電圧VDDLを動作電圧VDDHの1/2あたりに設定すると、前記の積が最大になる。

【0116】本発明をプロセッサを構成する回路に応用する場合には、上記基本構造に若干の変更を加えることにより更に消費電力を押さえることができる。即ち、プロセッサの制御ロジックでは、多くの場合、組み合わせ論理回路の主出力端子にラッチが設けられている。このような場合には、レベルコンバータとして図24に示す回路を使う代わりに、図20に示したようなレベルコンバータとラッチをミックスした「レベル変換機能付ラッチ」を使うことができる。この回路は、信号をラッチする機能と、動作電圧VDDLを動作電圧VDDHに変換する機能とを有する。

【0117】このレベル変換機能付ラッチは、動作電圧VDDHで動作するラッチ回路LHを備えている。ラッチ回路は、互いの入出力ノードを逆方向に接続した一対のインバータI2、I3からなっている。そして、その出力は、やはり動作電圧VDDHで動作するインバータI4の入力に接続され、インバータI4の出力が主出力端子となっている。インバータの一方は、NチャネルトランジスタMN4によって接地レベルに接続され、他方はNチャネルトランジスタMN5によって接地レベルに接続されている。これらNチャネルトランジスタMN4、MN5は、動作電圧VDDLで動作し、やはり動作電圧VDDLで動作するゲートの出力信号によってオンオフするが、NチャネルトランジスタMN5にはインバータが設けられているので、一方がオンしたときは他方

がオフする。又、ラッチ回路LHとNチャネルトランジスタMN4、MN5の間には、クロック信号CLKに同期して、ラッチ回路LHへの信号の取込を行うNチャネルトランジスタMN6、MN7が設けられている。図20に示す回路は、通常のラッチ回路とほぼ同じ消費電力で、レベル変換機能とラッチ機能の両方を満たすものであり、消費電力低減化の観点から、効率の良い回路である。

【0118】言うまでもなく、図20に示すレベル変換機能付ラッチを動作電圧VDDLが供給されるゲートに使うには、図3に示す回路構造、すなわち、動作電圧VDDLが供給されるゲートは主出力端子側に設けられている必要がある。なぜなら、レベル変換機能付ラッチは、元々ラッチが設けられているところにしか使えないからである。即ち、レベル変換機能付ラッチは、通常のレベルコンバータの代わりにはならない。このように、図3に提案する回路構造は、図20に示す低消費電力のレベル変換機能付ラッチ回路の利用に非常に都合のよいものである。

【0119】更に図3に示す回路構造は、パターンレイアウトの観点からしても、非常に利点が多い。即ち、動作電圧VDDHが供給されるゲートと動作電圧VDDLが供給されるゲートとが、それぞれ別々にクラスタ化されているので、シンプルなパターンレイアウトが実現する。

【0120】その様なレイアウトの具体的な例を、図21に示す。ここでは、スタンダード・セルを用いて、レイアウトをデザインする場合について説明をする。スタンダード・セルは、複数の列に並べられ、夫々の論理回路が列の内部に配置され、夫々の列を貫いて電源線が通っている。又、スタンダード・セル間の接続は隣接する列の間のチャンネルと呼ばれるスペースに設けられた配線で行われる。

【0121】図では、6つの列に図3の回路が実現されている。第1の列R1及び第2の列R2に動作電圧VDDHクラスタが組み立てられ、第3の列R3、第4の列R4及び第5の列R5に動作電圧VDDLクラスタが組み立てられ、第6の列R6にレベルコンバータが組み立てられている。この場合、主入力端子は第1の列R1に上端に位置し、主出力端子は第2の列R2の下端に位置する。従って、信号の流れは上から下へとむかうシンプルな回路レイアウトとなる。

【0122】図21に示すパターンレイアウト方式は、2つの理由から面積を小さくする上で有利である。その1つは、レベルコンバータの形成されている第6の列R6を除いて、スタンダード・セルの各列には、動作電圧VDDH又は動作電圧VDDLの何れか一方のみを供給すればよく、電源ラインは1つの列に対して1本で足りる。従って、2本の電源ラインを利用するスタンダード・セルを用いる場合に比較して、電源ラインの分だけ高

さ方向のサイズが小さくなる。

【0123】もう1つは、ウエルの分離が容易になるということである。もし、動作電圧VDDH及び動作電圧VDDLの2電源が共存するCMOS回路をレイアウトする際に、スタンダード・セルの同一列内に動作電圧VDDHが供給されるゲートと、動作電圧VDDLが供給されるゲートを配置する場合には、構成要素であるトランジスタは、次のように形成される。CMOS回路では、動作電圧VDDHが供給されるゲートのPチャネルトランジスタは、通常ウエル電位として動作電圧VDDHが与えられたNウエル内に形成される。一方、動作電圧VDDLが供給されるゲートのPチャネルトランジスタは、通常ウエル電位として動作電圧VDDLが与えられたNウエル内に形成される。この場合、ウエルの電位が異なるので、これら2種類のPチャネルトランジスタは、隣り合わせにおく場合、分離したNウエルに形成する必要がある。

【0124】これに対して、図21に示すパターンレイアウト方式では、スタンダード・セルの1つの列に形成されるトランジスタは、動作電圧VDDH又は動作電圧VDDLの何れか一方のみで動作し、ウエル電位は同一なので、ウエルの分離は一切必要ない。従って、ウエルの分離に必要な面積の分だけ、占有面積は小さくなる。

【0125】又、このようなレイアウトの、横方向(row方向)の配置では、信号経路がなるべく垂直方向に並べるようにして、信号線の長さを短くする配慮を行う。これにより、信号遅延が小さくなり、信号線の占めるスペースも少なくなる。

【0126】更に、図21に示すパターンレイアウトは、スタンダードセルであるものとしたが、ゲートアレイでも全く同様である。ゲートアレイでは、横方向ではなく縦方向をrow方向と呼ぶのが普通である。

【0127】上記、発明の実施形態では、2つの異なる電位の動作電圧を使用する場合について、具体的に説明した。3種類以上の動作電圧(例えば、2.5V、3.3V及び5V)を使う場合にも、本提案の構造は容易に応用できる。すなわち、主入力端子側には最も高い動作電圧で動かすゲートをクラスタ化し、さらに、そのクラスタの出力に、その次に高い動作電圧で動かすゲートをクラスタ化して接続し、以下同様に次々に接続して、最後に、最も主出力端子に近い側に最も低い動作電圧で動かすゲートをクラスタ化して接続すればよい。

【0128】またさらに、与えられた論理回路の外部に、該論理回路とは別に動作電圧VDDLで動かす回路があり、与えられた論理回路の出力を該外部回路で受けるような場合には、わざわざレベルコンバータで信号振幅を0～動作電圧VDDHまで増幅して出力する必要はない。このような場合には、該出力信号に対しては、与えられた論理回路内でレベルコンバータを挿入せずに、動作電圧VDDLクラスタ内のゲートから直接、論理回

路外部へ出力すればよい。

【0129】

【発明の効果】以上詳細に説明したように本発明によれば、レベルコンバータの個数を、最大でも出力ピンと同数に抑えることができるため、大規模半導体集積回路の消費電力の低減化の効果が非常に大きい。

【図面の簡単な説明】

【図1】図1は、本発明を利用して設計されたゲートアレイやスタンダードセル等のASICチップの全体を示す図である。

【図2】本発明による設計方法に適したランダムロジック部における信号バスとフリップフロップとの配置を示す図である。

【図3】本発明の実施形態により設計され、動作電圧VDDHのクラスタと動作電圧VDDLのクラスタからなるCMOS組み合わせ論理回路を示すブロックダイアグラムである。

【図4】各ゲートと、そのゲートが既に探索されているか否かを示すフィールド“Visited”、及びそのゲートに与えられている供給電圧レベル“L/H”を対応づけるテーブルを示す図である。

【図5】Depth-First-Searchに従って、動作電圧VDDHを動作電圧VDDLに置き換える処理のメインのルーチンを示すフローチャートである。

【図6】Depth-First-Searchに従って、動作電圧VDDHを動作電圧VDDLに置き換える処理のサブルーチンを示すフローチャートである。

【図7】各ゲートと、そのゲートが既に探索されているか否かを示すフィールド“Visited”、及びそのゲートに与えられている供給電圧レベル“L/M/H”を対応づけるテーブルを示す図である。

【図8】Depth-First-Searchに従って、動作電圧VDDHを動作電圧VDDL又は動作電圧VDDMに置き換える処理のメインのルーチンを示すフローチャートである。

【図9】Depth-First-Searchに従って、動作電圧VDDHを動作電圧VDDMに置き換える処理を示すフローチャートである。

【図10】Depth-First-Searchに従って、動作電圧VDDHを動作電圧VDDMに置き換える処理のサブルーチンを示すフローチャートである。

【図11】Depth-First-Searchに従って、動作電圧VDDHを動作電圧VDDL又は動作電圧VDDMに置き換える処理のサブルーチンを示すフローチャートである。

【図12】本発明の実施形態により設計され、動作電圧VDDHのクラスタ、動作電圧VDDLのクラスタ及び動作電圧VDDMのクラスタからなるCMOS組み合わせ論理回路を示すブロックダイアグラムである。

【図13】Bredth-First-Searchに従って、動作電圧VDDHを動作電圧VDDLに置き換える処理のメインの

ルーチンを示すフローチャートである。

【図14】Bredth-First-Searchに従って、動作電圧VDDHを動作電圧VDDLに置き換える処理のサブルーチンを示すフローチャートである。

【図15】Bredth-First-Searchに従って、動作電圧VDDHを動作電圧VDDL又は動作電圧VDDMに置き換える処理のメインのルーチンを示すフローチャートである。

【図16】Bredth-First-Searchに従って、動作電圧VDDHを動作電圧VDDMに置き換える処理を示すフローチャートである。

【図17】Bredth-First-Searchに従って、動作電圧VDDHを動作電圧VDDMに置き換える処理のサブルーチンを示すフローチャートである。

【図18】Bredth-First-Searchに従って、動作電圧VDDHを動作電圧VDDL又は動作電圧VDDMに置き換える処理のサブルーチンを示すフローチャートである。

【図19】低動作電圧VDDLの設定値と回路全体の消費電力との関係を示す図である。

【図20】信号をラッチする機能と、動作電圧VDDLを動作電圧VDDHに変換する機能とを有するレベル変換機能付ラッチを示す図である。

【図21】スタンダード・セルによって設計された本発明の実施形態によるCMOS組み合わせ論理回路を示すブロックダイアグラムである。

【図22】本発明が適用可能なCMOS組み合わせ論理回路を示すブロックダイアグラムである。

【図23】低動作電圧VDDLで動かす回路と高動作電圧VDDHで動かす回路をダイレクトに接続する場合に生じる問題点を示す図である。

【図24】レベルコンバータを示す回路図である。

【図25】消費電力を低減する為に、構成要素のゲートの一部に加わる動作電圧を下げる場合に生じる問題点を示す回路図である。

【符号の説明】

G1～G10 ゲート

LC レベルコンバータ

i1～i4 主入力端子

o1～o4 主出力端子

VDDH 高動作電圧

VDDL 低動作電圧

R1～R6 スタンダード・セル列

MP1、MP2、MP3 Pチャネルトランジスタ

MN1～MN7 Nチャネルトランジスタ

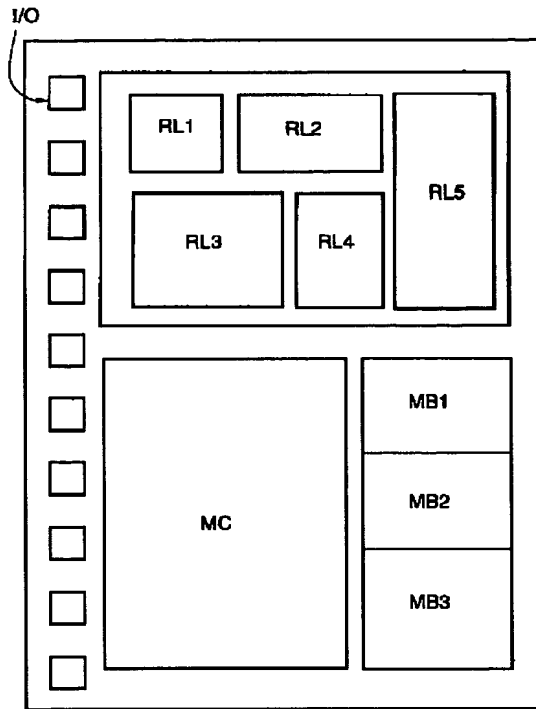
I1、I2、I3 インバータ

MC、MB1～MB3 メガセルで構成される機能ブロック

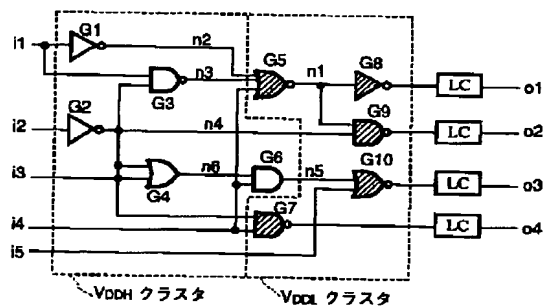
RL1～RL5 ランダムロジックを構成する機能ブロック

MC メモリ

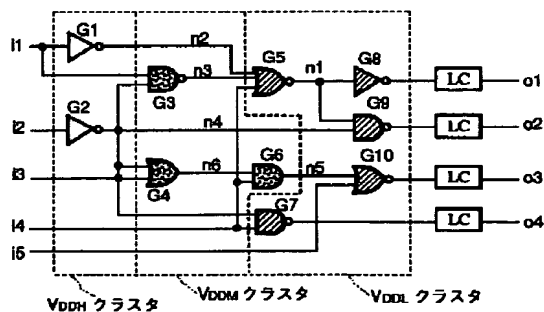
【図1】



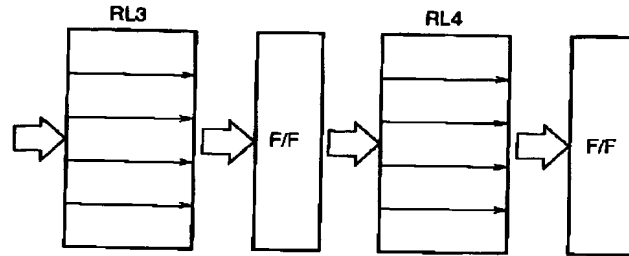
【図3】



【図12】



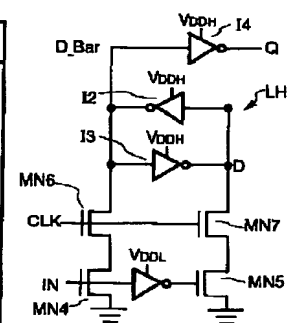
【図2】



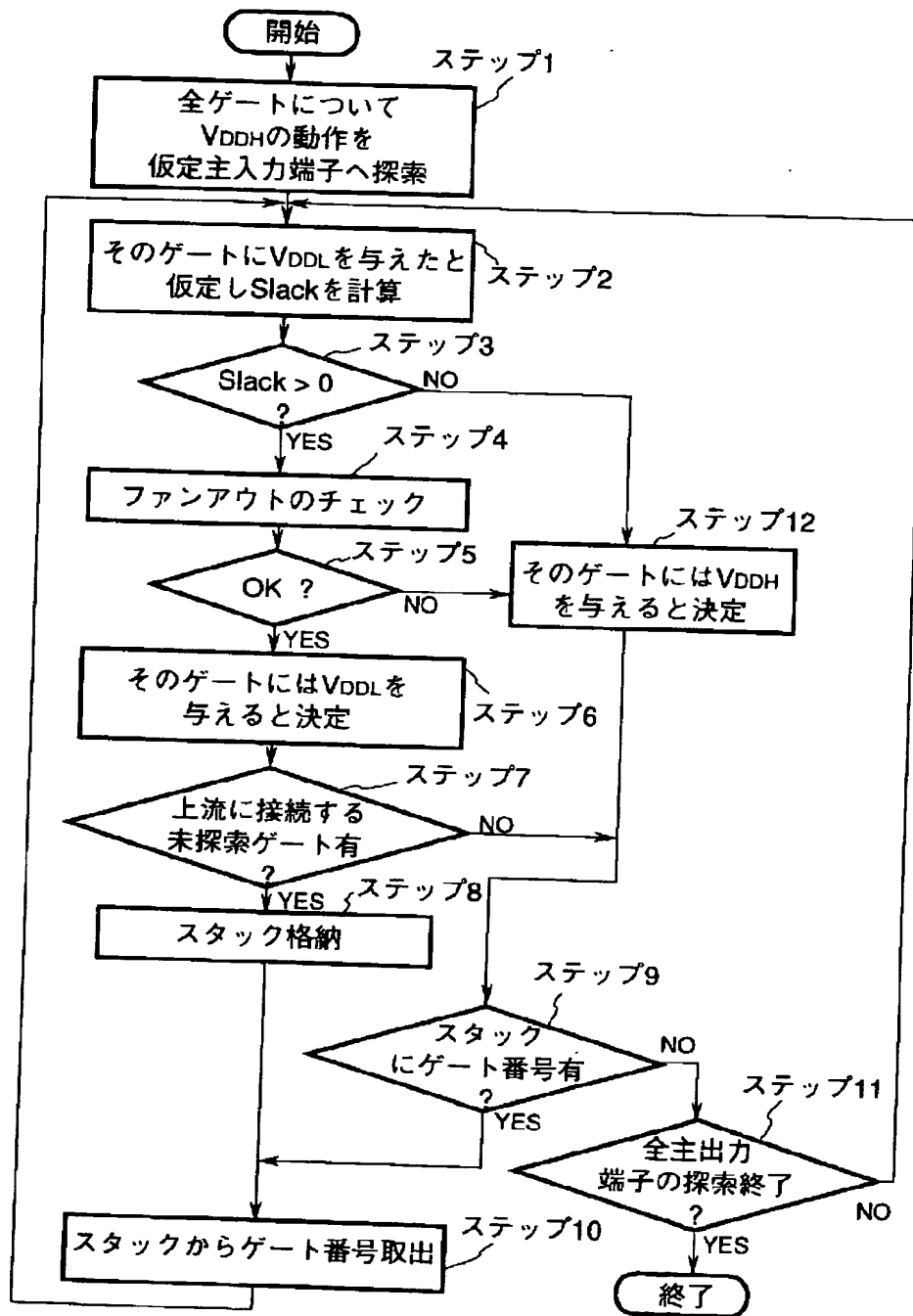
【図4】

Gate	Visited	L/H
G1	Yes	L
G2	Yes	L
G3	Yes	L
G4	Yes	H
G5	No	H
...

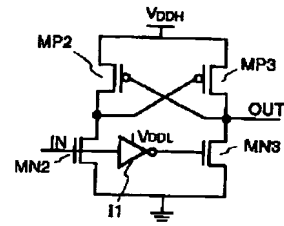
【図20】



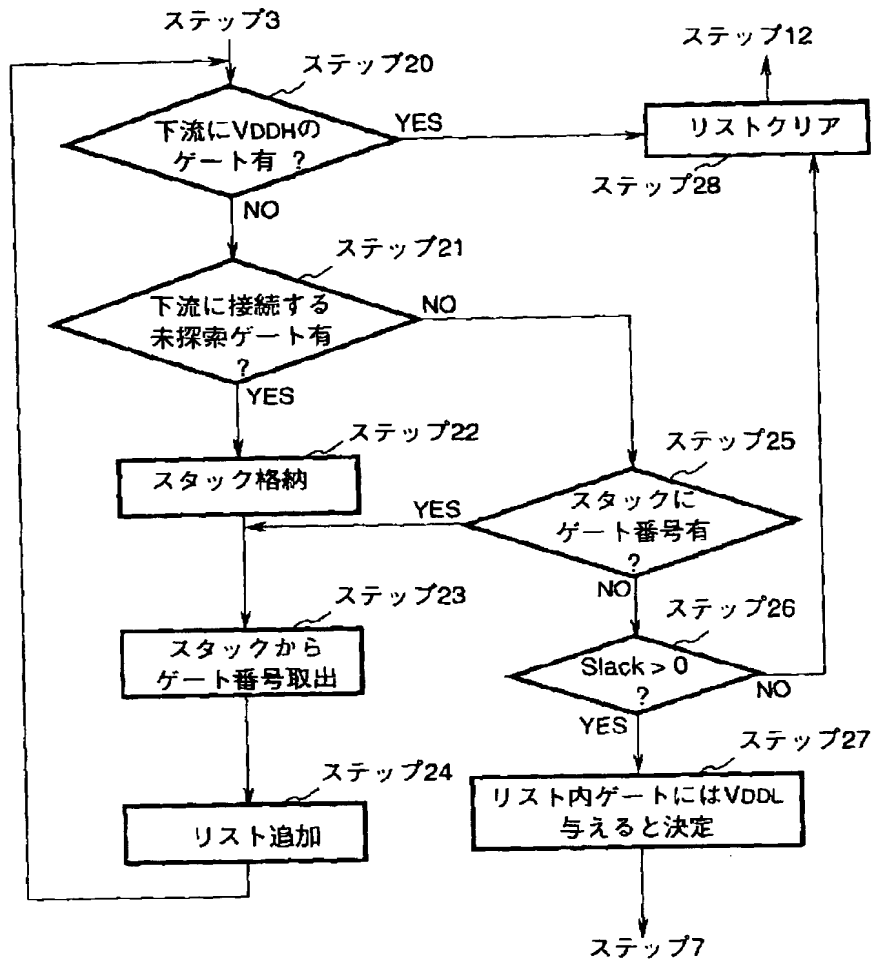
【図5】



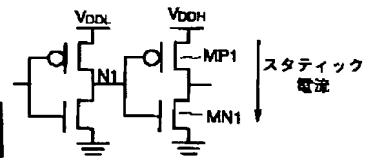
【図24】



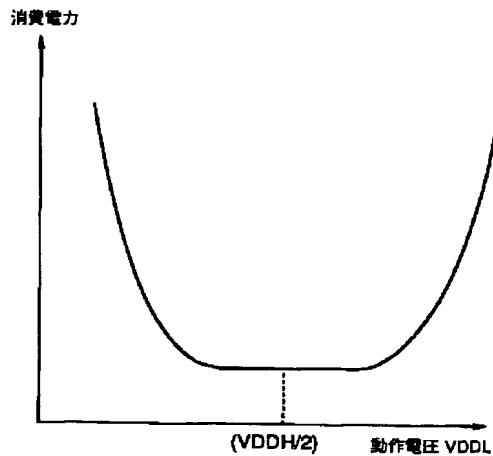
【図6】



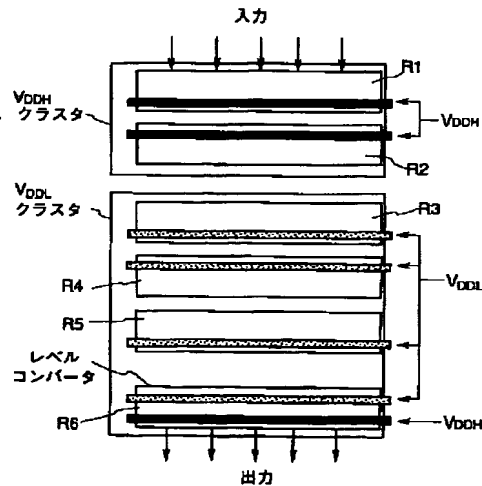
【図23】



【図19】



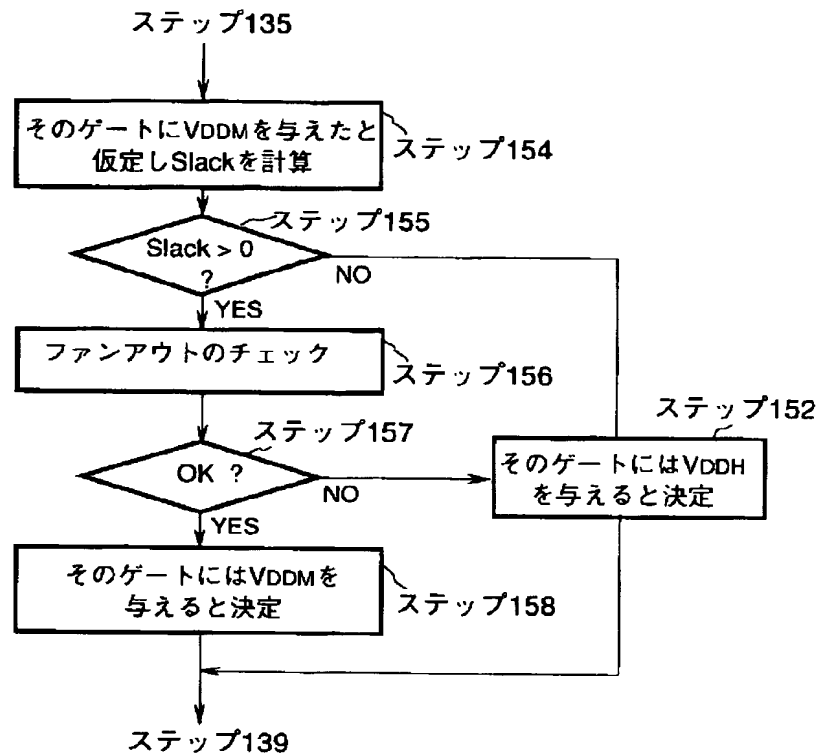
【図21】



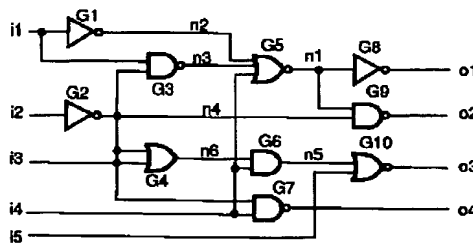
【図7】

Gate	Visited	L/M/H
G1	Yes	L
G2	Yes	L
G3	Yes	M
G4	Yes	H
G5	No	H
...

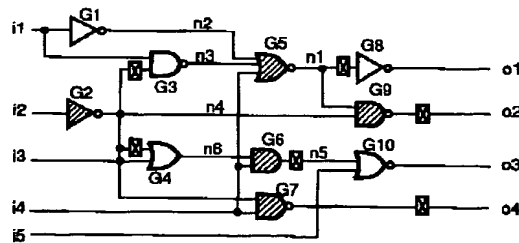
【図16】



【図22】



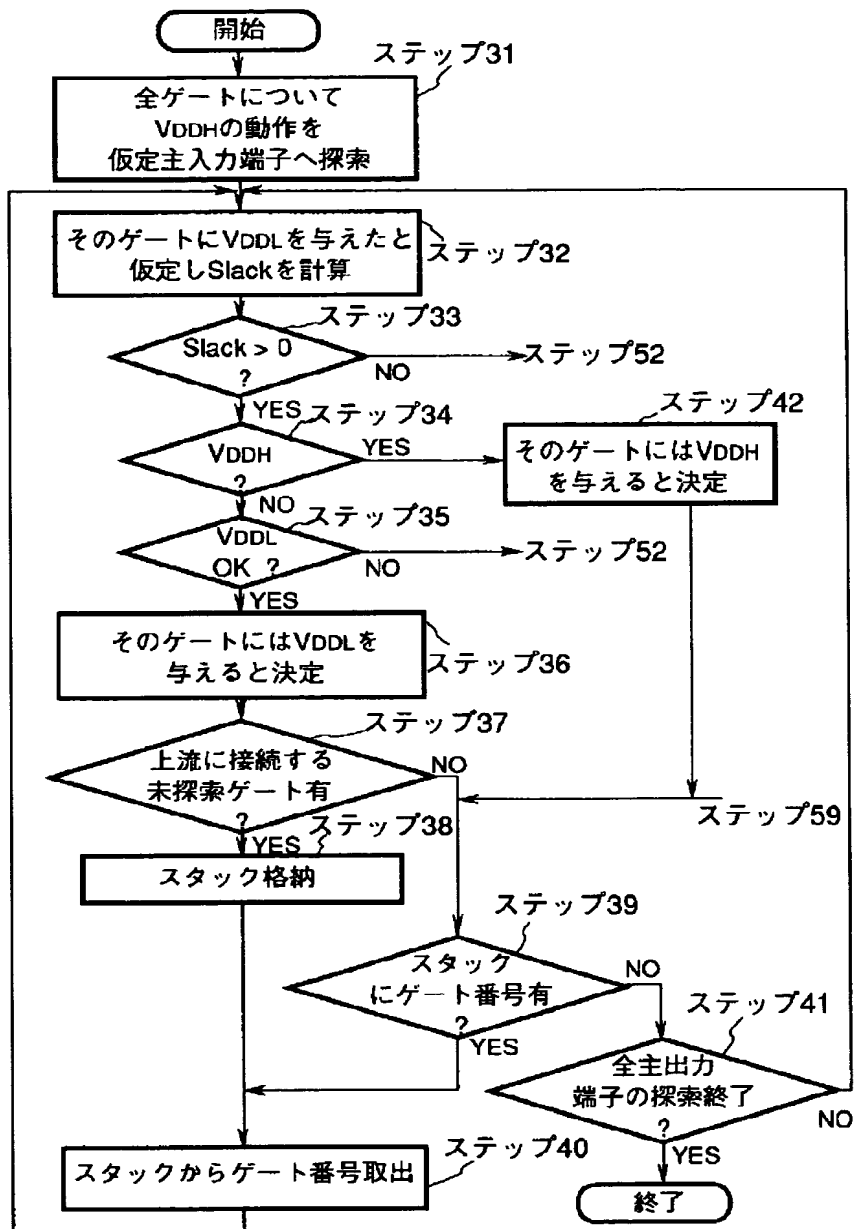
【図25】



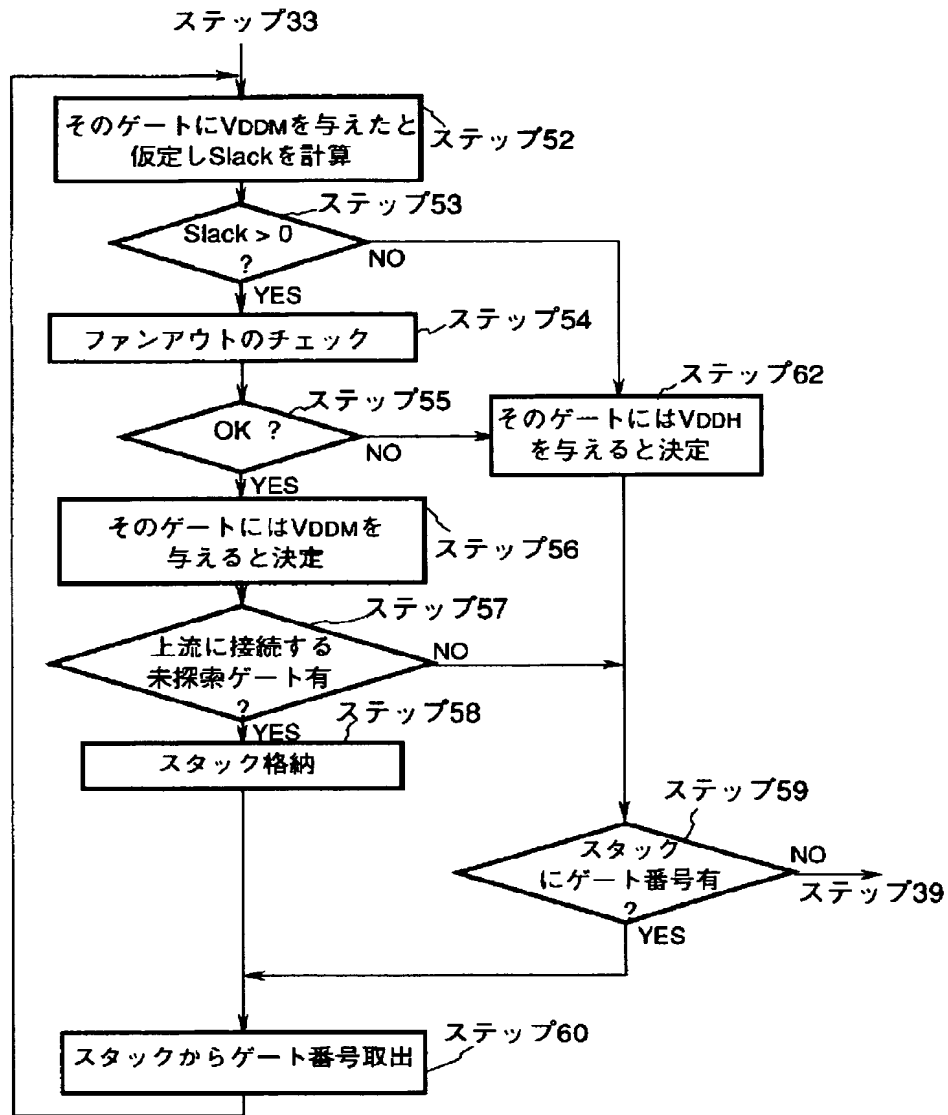
▨ VDDL を供給するゲート

▨ レベルコンバータ

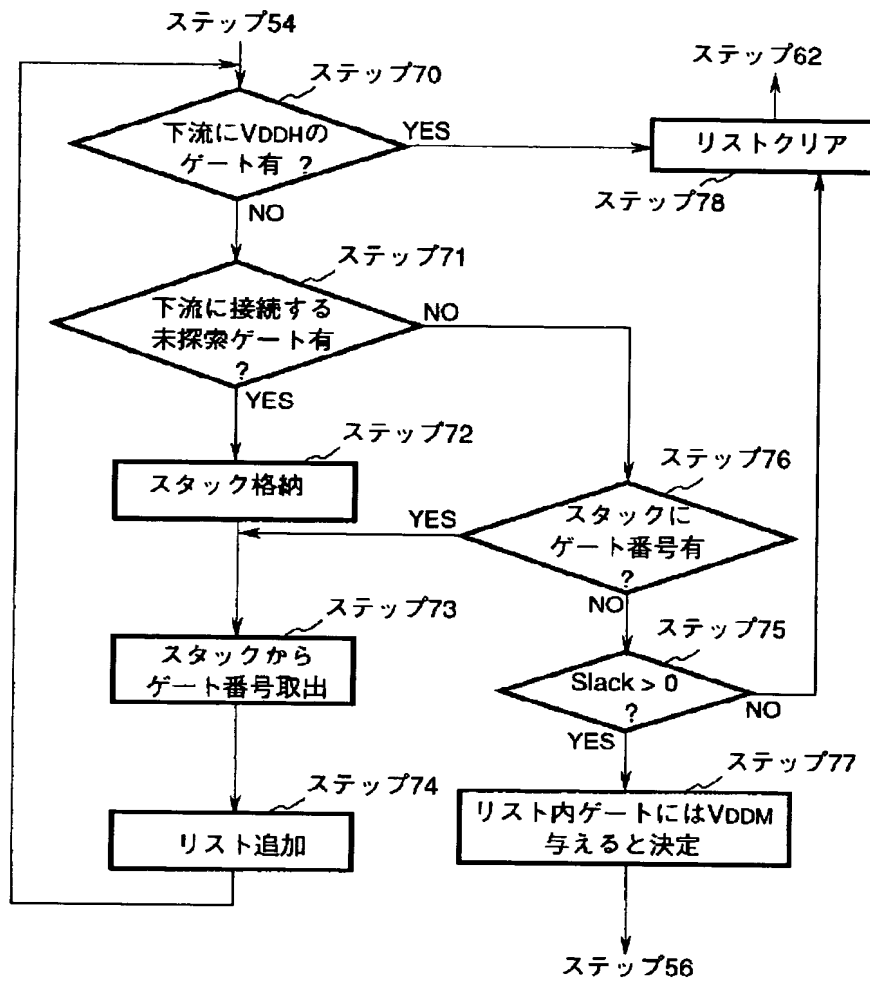
【図8】



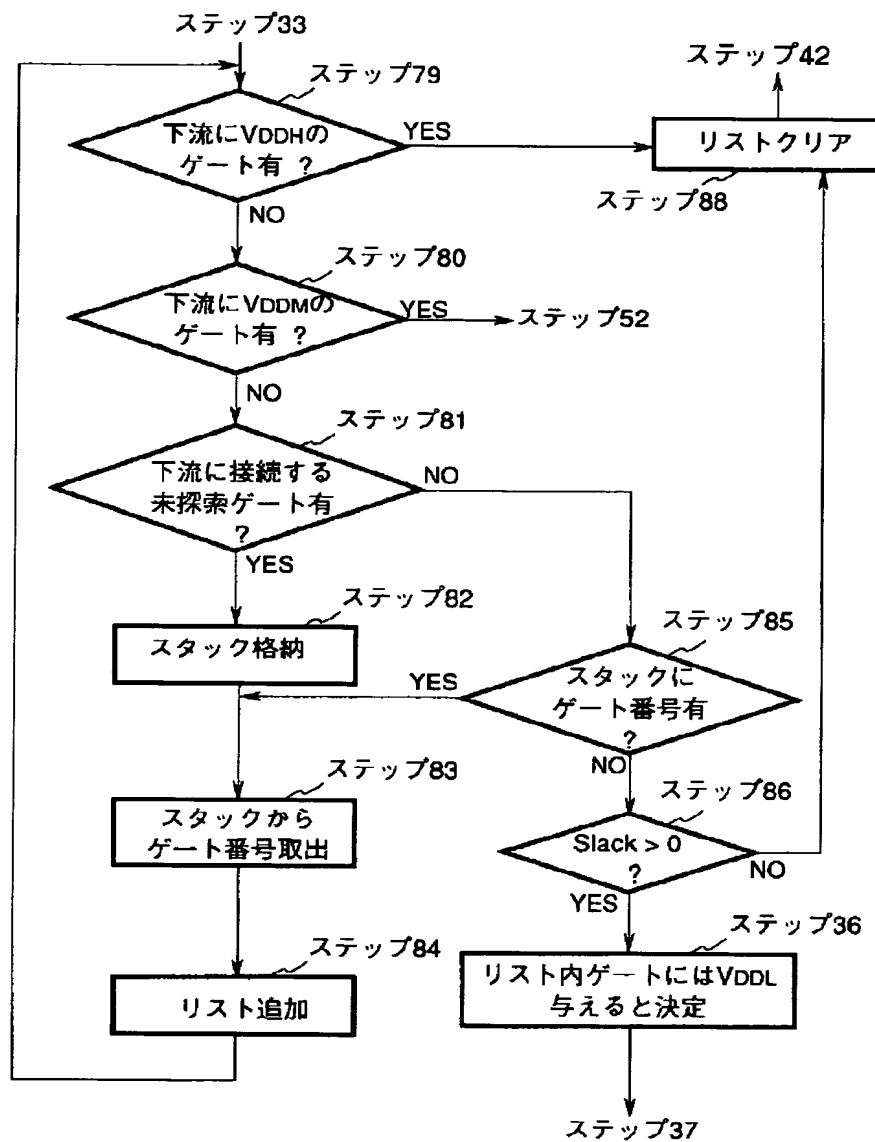
【図9】



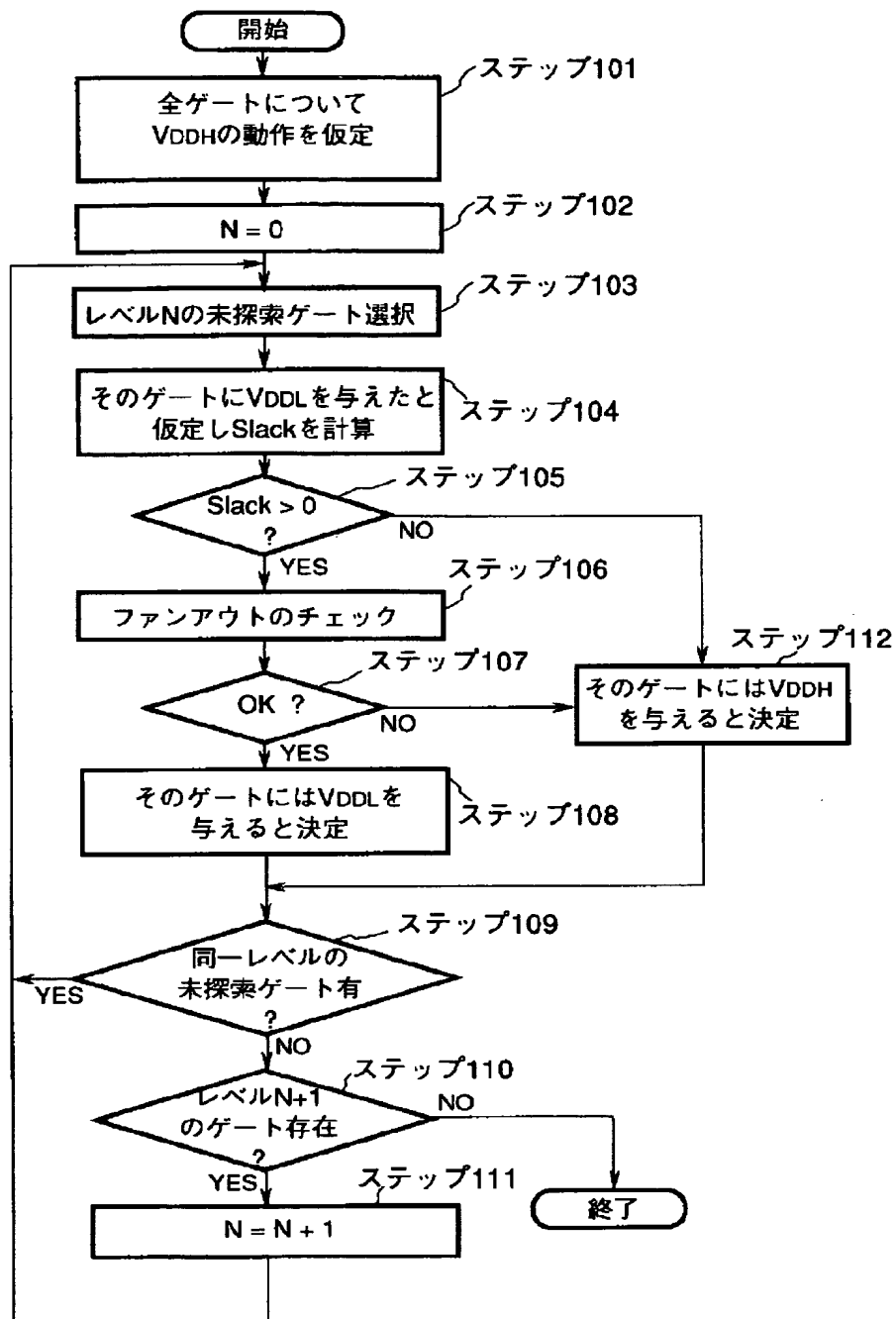
【図10】



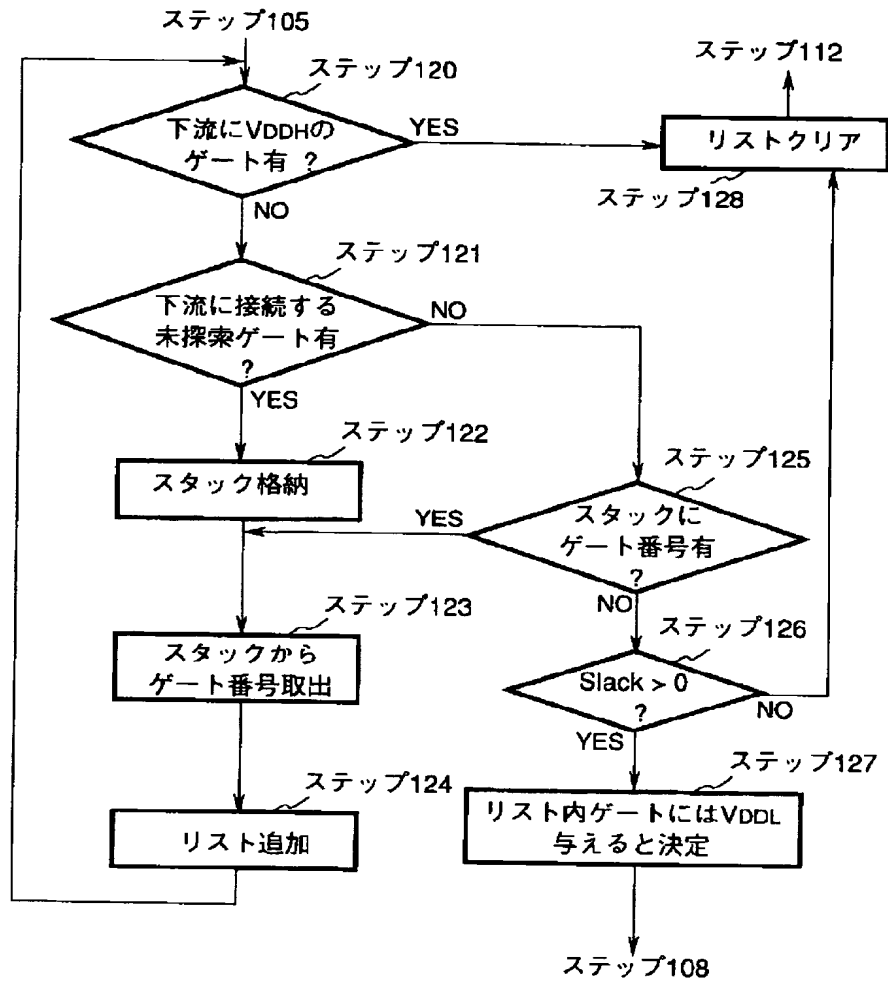
【図11】



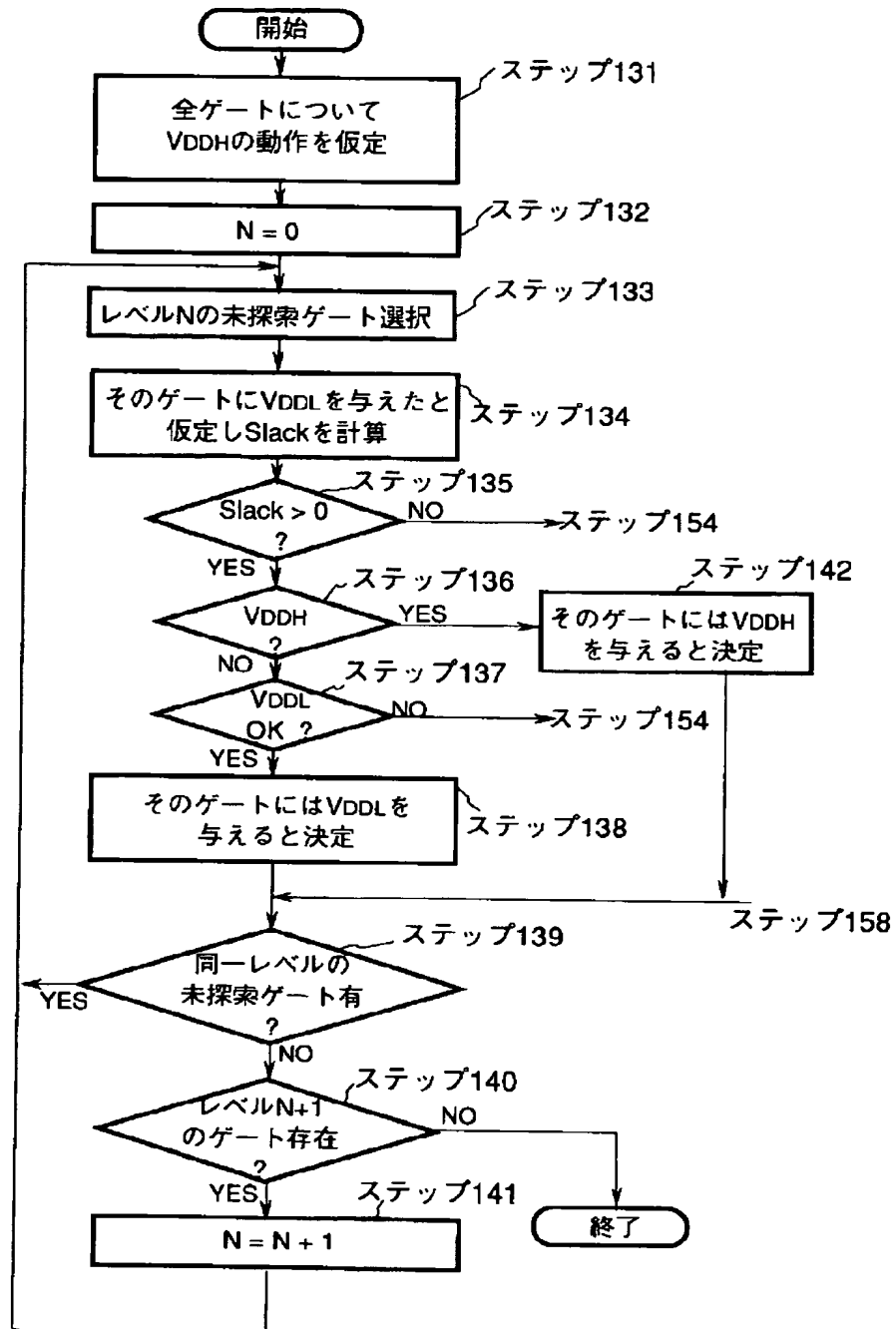
【図13】



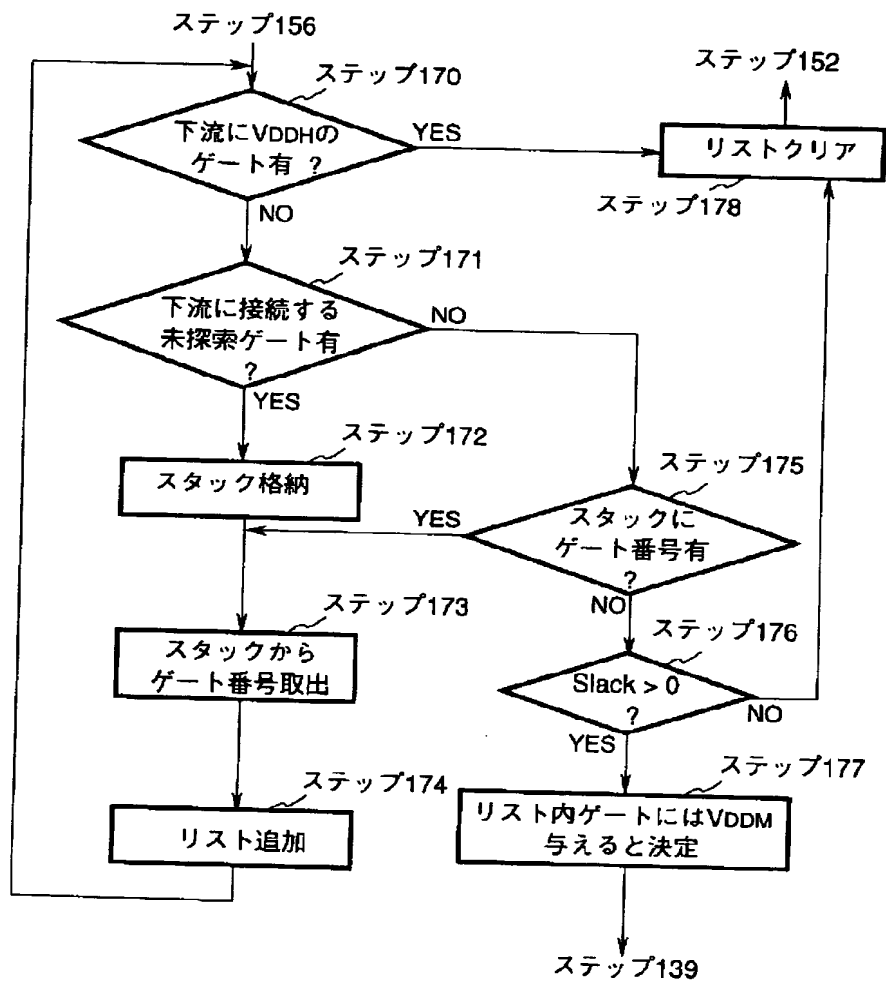
【図14】



【図15】



【図17】



【図18】

